Predictive Analytics for Maintaining Power System Stability in Smart Energy Communities

by

Aida Mehdipour Pirbazari

A dissertation submitted in partial satisfaction of
the requirements for the degree
PHILOSOPHIAE DOCTOR (PhD)

University of
Stavanger

Faculty of Science and Technology
Department of Electrical Engineering and Computer Science
February 2021

University of Stavanger
N-4036 Stavanger
NORWAY
www.uis.no

ISBN 978-82-8439-005-5
ISSN 1890-1387

PhD Thesis UiS no. 587

Acknowledgements

I would like to express my sincere gratitude to my esteemed supervisor, Professor Chunming Rong, for his dedicated support and supervision during this research. I am grateful to him for his patience, encouragement, and insightful suggestions that accompanied me in this study.

I am also thankful to my Co. supervisor Assoc. Prof. Antorweep Chakravorty for his thoughtful comments and recommendations on this dissertation. He was always willing and enthusiastic to assist during my research work.

My sincere thanks also go to Prof. Wilfried Elmenreich and Ekanki Mohan, Institute of Networked and Embedded Systems, University of Klagenfurt, Austria, who provided me with an opportunity to join their team and extend my analysis from various perspectives.

I gratefully acknowledge the financial and consultancy supports received towards my PhD from the Research Council of Norway, International Research Institute of Stavanger (IRIS) as well as Lyse, DNV.GL and Statnett as system operators of the Norwegian power system.

I would also like to thank my dear friends and colleagues at the UiS; Mina Farmanbar, Jayachander Surbiryala, Albana Rochi, Faraz Barzideh, Cristina Viorica Hegedus, Nikita Rajendra Karandikar and Dhanya Therese Jose for their valuable support and friendship over the course of this study.

Special thanks to my parents, brother and sister for supporting me spiritually throughout this project, and in my whole life. To my dear sons Barsam and Behnood – thanks for allowing me time away from you to do the work. Finally many thanks to my caring and supporting husband, Ali for his patience, encouragement and understanding during my PhD. The completion of this thesis could not have been possible without them.

Aida Mehdipour Pirbazari, March 2021

Preface

This dissertation is submitted in partial fulfillment of the requirement for the degree of Philosophiae Doctor (PhD) at the University of Stavanger, Norway. The study was carried out during the period from November 2017 to February 2021. The dissertation is written on the basis of the published and accepted research papers. The articles are reformatted to fit the thesis's structure. The contents of the original articles are self-contained.

Abstract

Digitalization and decentralization of energy supply have introduced several challenges to emerging power grids known as smart grids. One of the significant challenges, on the demand side, is preserving the stability of the power systems due to locally distributed energy sources such as micro-power generation and storage units among energy prosumers at the household and community levels. In this context, energy prosumers are defined as energy consumers who also generate, store and trade energy. Accurate predictions of energy supply and electric demand of prosuemrs can address the stability issues at local levels. This study aims to develop appropriate forecasting frameworks for such environments to preserve power stability.

Building on existing work on energy forecasting at low-aggregated levels, it asks: What factors influence most on consumption and generation patterns of residential customers as energy prosumers. It also investigates how the accuracy of forecasting models at the household and community levels can be improved.

Based on a review of the literature on energy forecasting and performing empirical study on real datasets, the forecasting frameworks were developed focusing on short-term prediction horizons. These frameworks are built upon predictive analytics including data collection, data analysis, data preprocessing, and predictive machine learning algorithms based on statistical learning, artificial neural networks and deep learning.

Analysis of experimental results demonstrated that load observations from previous hours (lagged loads) along with air temperature and time variables highly affects the households' consumption and generation behaviour. The results also indicate that the prediction accuracy of adopted machine learning techniques can be improved by feeding them with highly influential variables and appliance-level data as well as by combining multiple learning algorithms ranging from conventional to deep neural networks. Further research is needed to investigate online approaches that could strengthen the effectiveness of forecasting in time-sensitive energy environments.

Contents

List of Tables

List of Figures

List of Papers

The following papers are included in this thesis:

- **Paper I**

  Evaluating Feature Selection Methods for Short-Term Load Forecasting
  A.M. Pirbazari, A. Chakravorty, C. Rong
  Published in the proceedings of 6.th IEEE International Conference on Big Data and Smart Computing (BigComp) 2019.

- **Paper II**

  Short-Term Load Forecasting Using Smart Meter Data: A Generalization Analysis
  A.M. Pirbazari, M. Farmanbar, A. Chakravorty, C. Rong
  Published in the Processes Journal, belonging to special issue: Clean Energy Conversion Processes 2020.

- **Paper III**

  Improving Load Forecast Accuracy of Households Using Load Disaggregation Techniques 2020.
  A.M. Pirbazari, M. Farmanbar, A. Chakravorty, C. Rong

Published in the proceedings of 6.th IEEE International Conferences on Smart Data (SmartData) 2020.

- **Paper IV**

An Ensemble Approach for Multi-step Ahead Energy Forecasting of Household Communities
A.M. Pirbazari, E. Mohan, A. Chakravorty, W. Elmenreich, C. Rong
Accepted for publication in IEEE Access Journal 2021.

Chapter 1

Introduction

This chapter provides an introduction to the research work and it is structured as follows. The first section shortly define the main concepts of the research. Section 2 describes the research problem and the motivation behind the research work. The third section presents the main objective and research questions. Section 4 lists the research articles followed by Section 5 that provides the outline of the thesis.

1.1 Definitions

1.1.1 Predictive Analytics

Predictive analytics is a branch of advanced analytics to make predictions about future or unknown events based on current and historical facts. The process of predictive analytics includes six steps: (1) project definition which identifies the project outcomes, business objectives and required data sources. (2) Data collection and data mining that provide data from various sources for analysis. (3) Data analysis aimed at cleaning and transforming data into useful information. (4) Statistical analysis allows assumptions and conclusions to be tested and evaluated using a standard statistical model. (5) Predictive modelling that enables the development of predictive models using regression, machine learning and artificial intelligence. (6) Deployment which allows the deployment of forecasts into the everyday decision-making process. During deployment, the models are validated, scored or integrated with reporting or business applications. The deployed models are continuously monitored for maintaining and improving performance [1]. There are different industrial applications of predictive analytics such as insurance, banking, healthcare, marketing and energy industries such as oil and gas, electricity, etc.

1.1.2 Conventional Power Grid

In a traditional power grid, there is a one-way flow of electricity from power generators to consumers. The electricity generation occurs at centralized facilities such as steam stations and fossil-fuel-fired power plants and is further distributed through the long-distance high-voltage transmission lines to multiple end-users.

The conventional electricity grid has been constantly upgraded with new technologies, including higher voltage equipment, advanced power electronics, digitalization of control mechanisms and demand response programs [2]. Nevertheless, there are still major issues with the existent electricity infrastructure. One is the efficiency of transmission lines. The different countries experience various amounts of electricity loss in the transmission and distribution networks. The studies show that in 2016 [3], the total amount of electricity loss among developing countries ranged between 16% and 50% where power is mostly transmitted over long distances to several dispersed rural areas. However, losses are recorded lower in more developed countries with more effective transmitting networks, e.g., the United States and Germany suffered only 6% and 5% losses, respectively.

The next concerns are about the reliability and security of the network. Any failure or disruption in the power supply due to ageing infrastructure, natural disasters or cyber-attacks can quickly spread and significantly disrupt the power grid. For example, in December 2015, a massive cyberattack occurred in Ukraine's power system leading to a long-term power outage across houses and facilities. In 2021, the Texas power grid experienced its worst blackouts for the decades caused by the winter storm.

Furthermore, there are always environmental impacts associated with electricity system specifically with centralized power generation.

Air pollution from burning fuels such as coal and natural gas, water usage for steam production, solid and often toxic waste as the effects of power generation, as well as land use for major power plant operations are among the environmental issues which drive a transition to a greener grid [4].

1.1.3 Smart Grid and Micro Grid

To address the issues mentioned above, the power system is shifting towards the modern two-way power flow system known as a smart grid. In this new environment, the interactions between different components of the grid are facilitated through information and communication technologies (ICTs). The smart grid incorporates a wide range of operating and energy measures such as smart meters, distributed and renewable energy infrastructure as well as intelligent energy management mechanisms in order to optimize the use of installed infrastructure, improve security, enhance power quality, and mitigate costly environmental effects.

To meet power demand in local regions, microgrids have been evolved. A microgrid as a part of the smart grid is a local electricity infrastructure that covers a particular local area such as a hospital, a university campus or neighbourhood. To satisfy its power supply needs, it combines a range of distributed energy technologies such as renewable energy, integrated heating and power, and energy storage systems. The goal of such a system is achieving a green, reliable and cost-effective local network by providing stable energy from distributed resources that are increasingly renewable and affordable [5].

1.1.4 Smart Energy Communities

Technical developments along with reduced costs in micro-generators and energy storage devices; enable end-users of electricity to become prosumers. The word ‘prosumers’ coined by Alvin Toffler in 1970 [6] refers to the consumers who would become producers. In the electric power industry, prosumers are the electricity customers who can contribute to the energy supply by locally generating, storing and selling electricity in their domestic environments. As compared to the traditional grid, where end-users merely purchase electricity from retailers, the prosumers can produce their electricity from micro-scale renewable energy generation units such as photovoltaic solar panels and small wind turbines. The surplus generation furthermore can be either stored in batteries or sold to the grid via various tariff schemes.

A smart energy community is formed when a group of prosumers produce green energy, trade micro-production or sell it to the main grid. A community of prosumers such as residential customers can better manage their electrical needs through generation from renewable energy sources, battery storage systems and microgrids. Microgrids through advanced software and control systems facilitate local supply and trade of energy for the community members. This would reduce the dependency of the smart community from the centralized power grid thus leading to the reduction in electricity transfer capacity to and from the main grid as well as the increase in reliability of supply and self-sufficiency [7]. The community's members can also cooperate and interact with each other through an intelligent component called a community gateway. The gateway is responsible for connecting the main grid to smart controlling devices within the prosumer community. It mainly facilitates local supply and trade of energy for the community members [8].

1.2 Problem Description and Motivation

The sustainable integration of smart energy communities with the main grid introduces multiple challenges to the management of microgrids. The widespread use of volatile renewable energy and the integration of highly complex loads within energy communities can disrupt the balance between supply and demand [9]. Technical problems such as power fluctuations, harmonics, as well as voltage and frequency fluctuations can occur during contact with the main grid. Each can affect the power supply's long-term and short-term stability [10].

Given the above circumstances, forecasting the energy generation and load demand of prosumers is essential in reducing the uncertainties caused by the integration to the grid as well as interactions between community members. Accurate load forecasting can provide the regional microgrid with the opportunity to balance supply and demand in both the short and long terms. Predicting peak consumption and microgeneration would enable transmission and distribution system operators to create an intelligent battery management system to determine when to use batteries instead of the grid when to share power with the grid and emergency backups [11].

In recent decades, a broad variety of research has discussed the issue of load forecasting at low aggregation levels such as substations, communities and buildings. However, it remains a complex problem for multiple reasons specifically for residential loads. First, load consumption in nature is a time-series data whose value at a present time has a very complex correlation with its value in previous times [12]. Specifically, at an individual building level such as a household, it exhibits several levels of seasonality, e.g. load at a given hour is not only dependent on the previous hour but also the load at the same hour on the previous day [13]. Second, many fluctuating factors affect the energy consumption of a residential building with different degrees such as weather conditions, the parameters relating to the house construction and consumption behaviour of the households [14]. Third, multiple variables in a smart household community, such as realistic demand dynamics, real-time data and decentralized energy sources, impact the precision of load forecasting thus requiring more sophisticated and nuanced forecasting models[11].

Accurate forecasting of renewable energy generation at a micro-scale, similar to load forecasting in the residential network, will be difficult for two reasons. First, in general, the energy data such as wind and solar energies are intermittent and chaotic due to their dependency on uncertain meteorological factors, such as solar irradiance, atmospheric temperature, module temperature, wind pressure and wind direction. Second, this volatility and unpredictability would become more complicated in the community microgrid where a variety of loads; Electric Vehicles and energy storage systems are incorporated in a more complex context.

There have been many research works related to the forecasting of renewable energy resources [15], [16], [17], [18] and [19]. However, the majority of research on solar energy is more focused on the estimation of solar radiation rather than the generation of solar power. Furthermore, a few studies have investigated the potential of predictive modelling for small-scale energy generation at local levels.

Up to our knowledge, also a few previous studies have considered the volatility of demand-side power and renewable energy output simultaneously in a community-based environment. Therefore, to enhance the reliability and promote the supply-demand balance in a prosumer community, it becomes necessary to devise forecasting methods with high effectiveness and efficiency for both energy loads.

1.3 Research Objective and Research Questions

The main objective of this thesis is to develop frameworks to provide energy forecasts at household and community levels with high accuracy and scalability through predictive analytics. The forecasting methods should have multiple characteristics to meet the requirements of the described environment.

First, they are required to be adaptive such that they can learn from data with limited human intervention since, explicit information about building construction or micro-generation units may not always be available. The models should also leverage modern techniques such as machine learning and Artificial Intelligence (AI) to address the complexity and temporal dependencies of energy data such as electricity, solar and wind. Besides, they must be scalable so that a large set of input data that would be collected from a vast number of sources can be processed efficiently. Finally, to extend the generalization ability, they need to be evaluated over different time horizons and consumption profiles.

The adoption of these frameworks would be useful to address the challenges in load balancing, power micro-generation and energy storage at smart energy communities. The deployment of final results, therefore, would meet the necessities of many actors in the energy market specifically transmission system operators, distributors and electricity companies. The research questions (RQs) proposed in the study are stated as follows:

(1) RQ 1. Which factors influence the most on consumption and generation behaviour of prosumers?

(2) RQ 2. What techniques are appropriate and highly accurate for prediction of energy consumption and small-scale energy generation of households?

(3) RQ 3. How can we improve the performance of successful predictive models at the building/local levels?

1.4 Research Publications

To answer the research questions, four experimental research are performed; three are published and one has been accepted for publication in a journal. All four research papers are included in the thesis. Fig.1 presents the relationship between appended papers and the research questions. Paper I [20] discusses different factors which influence most on various load consumption profiles at the household level, Paper II [21] studies and evaluates the performance of the most common algorithms in short-term load forecasting at the building level. Paper III [22] and Paper IV consider different approaches to improve short-term load forecasting at household and small community levels respectively. Paper IV also investigates the most influential factors on the micro-power generation of households.

An introduction to the research publications is provided below:

- Paper I [20]: "Evaluating Feature Selection Methods for Short-Term Load Forecasting", was published in Proceedings of IEEE International Conference on Big Data and Smart Computing (BigComp), 2019.

  In this paper, we have analyzed a set of candidate factors (features) which influence on energy consumption of different households with varying degrees of daily load volatility. We have also discussed and evaluated the importance of feature selection methods in improving the performance of forecasting models.

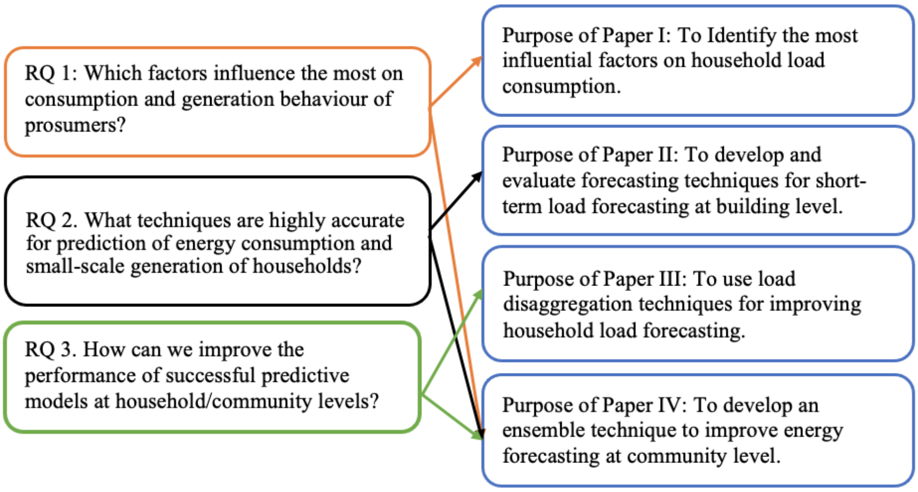

Figure 1.1: Relation between appended papers and research questions

- Paper II [21]: "Short-Term Load Forecasting Using Smart Meter Data: A Generalization Analysis", was published in Processes open access journal, belonging to special issue Clean Energy Conversion Processes, 2020, 8, 484.

  In this paper, we have developed and compared four predictive models based on machine learning algorithms to forecast daily peak and hourly energy consumption of residential buildings. We have considered a scenario where we only have access to buildings' historical load data (smart meter measurements) to build the forecasting models. We have also investigated the generalization ability of the models when they are evaluated on unseen house profiles during training.

- Paper III [22]: "Improving Load Forecast Accuracy of Households Using Load Disaggregation Techniques" was published in Proceedings of 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics).

  In this paper, we have proposed a hybrid approach to improve household load forecasting based on appliance-level data. The proposed approach enables the use of high-resolution smart meter data for hourly load forecasting by incorporating Non-Intrusive Load Monitoring (NILM) technique as a pre-processing step.

- Paper IV: “An ensemble approach for multi-step ahead energy forecasting of household communities” accepted for publication in IEEE Access journal, 2021.

  In this paper, we have analyzed various factors which influence on consumption and generation patterns of prosumers at an aggregated level. To improve short-term energy forecasting at this level, we have proposed a framework which utilizes an ensemble of deep recurrent neural networks and the most informative factors as advanced input to the models.

1.5 Thesis Outline

The remaining contents of this thesis are organized as follows. Chapter 2 provides an introduction to load consumption and renewable energy forecasting. It briefly explains their importance in the power grid and provides overviews of the existing forecasting techniques. Chapter 3 presents the background of technologies and machine learning techniques which are included in the research papers. A summary of four research publications is provided in Chapter 4; followed by Chapter 5 that concludes the thesis and discusses future work.

Chapter 2

Load and Renewable Energy Forecasting

This chapter describes the importance and applications of electrical load forecasting alongside renewable energy forecasting in the current and future power grid. It also briefly reviews the existing forecasting methods concerning energy consumption and generation of electricity customers.

2.1 Electric Load Forecasting

2.1.1 Definition

Electrical energy must be produced in response to consumer demand. It is thus important for energy providers to provide reliable forecasts of potential demand. Forecasting this load ahead of time is called load forecasting. Load demand forecasts are necessary for planning and setting generation capacity, transmission, and distribution needs.

2.1.2 Applications

For more than a century, electricity load predictions have played a critical role in the electricity industry [23]. Electric utilities need load forecasts for several business purposes based on different forecast horizons or time scales. Very short-term forecasts (from minutes to one hour ahead) are mostly applied to flow control, real-time grid operations and regulatory actions. Short-term forecasts (from one hour to several hours ahead) are typically used in economic load dispatch planning, load reasonable decisions and operational security in the electricity market. Medium-term (from several hours to several weeks or months ahead) predictions provide information to make decisions on unit commitment and reserve requirements. Finally, long-term forecasting (from several months to several years ahead) is normally used for maintenance planning, operation management, and feasibility study for the design of power infrastructures [24].

2.1.3 Growing Trend in Research Publications

Researchers have been studying load forecasting for decades, but due to the major shifts in the power grid, more researchers are drawn to the topic than ever before. For instance, as depicted in Fig.2.1 the growing trend in the number of research contributions on electric load forecasting, illustrates the importance of its application domain during the past two decades. We can also see that among published research papers, the contributions to short-term and very-short-term load forecasting have always had a significant share.

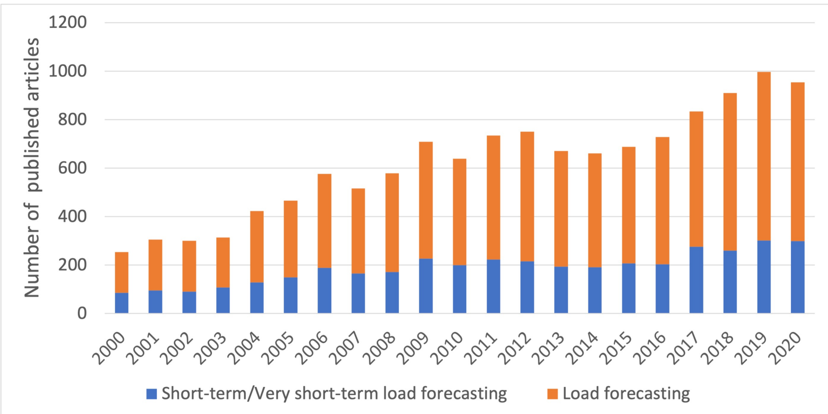

Figure 2.1: Global research trends in electricity load forecasting over two decades

Usually, one of the key factors behind the rising research developments in load demand forecasting could be the implementation of new power grid technologies and eventually the growth of microgrids, intelligent buildings. Smart meters, electric vehicles, solar batteries, solar panels are the most common samples of these technologies. The incorporation of all these in smart energy communities introduces new challenges to the power grid for preserving power stability. Therefore, the need for developing forecasting models to maintain equilibrium between demand and supply at lower aggregation levels such as buildings and communities has increased significantly.

2.1.4 Forecasting Methods

Since the 1970s, various load forecasting methods have been developed and proposed. The applied forecasting methods are mainly different depending on several criteria. The most important criteria are listed as follows.

2.1.4.1 Short-term vs Long-term

As described in the previous section, based on the duration of the forecast horizon, load forecasting approaches can be divided into four groups. Common methods known as 'trend analysis', 'end-use' and 'econometric' [25] are broadly used for medium and long-term forecasts. For short-term load forecasting, however, a number of methods such as similar day approach, regression and time series models, neural networks, fuzzy logic, and expert systems have been developed. As the emphasis of this study is on this category, the rest of this section will include a brief overview of strategies mainly applicable to short-term horizons.

2.1.4.2 High vs Low Aggregation Load Level

Forecasting methods have been applied in the areas with different geographical scales e.g. country, region, city, district and building. The forecasting at larger regions or units requires high aggregation of loads while forecasting at smaller areas needs lower load aggregation. The studies in [26] and [27] conclude that prediction task at smaller scales such as an individual building level can be more challenging than aggregate load forecasting. Since, for example, a country's load curve has a much smoother and more predictable profile than that of disaggregated environments like a residential building or a community. In this study, we focus on low-aggregation levels such as households or a group of households.

2.1.4.3 Uni-variate vs Multivariate

Regarding the input parameters to the predictive models, there have been several studies such as [28], [29] and [21] which investigated only one variable relating to load parameters (load profile, peak load, aggregate load, etc.) as the main contributing factors to the prediction; while, many researchers added other variables to the input vector such as weather conditions [30], calendar information [31] and customer behaviour [32]. An overall survey of different forecasting techniques [11] reveals that in scenarios where the horizon of forecast increases or when the aggregation level decreases, more parameters are usually added to the model, to capture the volatility of consumption patterns more precisely. This thesis investigates both input modelling: Uni-variate mainly in Paper II and Multivariate in Paper I [20], Paper III [22] and Paper IV.

2.1.4.4 One-step vs Multi-step Forecasting

One step forecasting estimates the target variable(s) one step ahead in time while multi-step forecasting predicts multiple time steps into the future. It is typically a simple task to predict chaotic time series one or a few time steps ahead, as shown by the high accuracy achieved in many systems, in both discrete and continuous time scales [33], [34]. When it comes to longer forecast horizons, the prediction task becomes more complicated due to the gradual growth of small forecasting errors resulting from the chaotic nature of real observations [35].

There are four main strategies for multi-step forecasting of time series data such as energy data: (1) training one model for each future time step. This strategy adds computational burden specifically with increasing the number of future time steps. Moreover, the in-dependency of trained models does not allow capturing the potential dependencies between the predictions. (2) Performing one-step forecasting multiple times in a recursive manner such that the forecast at the current time step is used as input for forecasting the next time step. This method would lead to accumulation of prediction errors as the forecast horizon increases. (3) Training separate models for each future time step such that each model is fed by the predictions made by models at previous time steps. This would typically overcome the limitations of the first and second strategies. (4) Training one model which can produce all the future time steps at once. Applying such an approach will require a complex and powerful learning model that, apart from between input and output variables, can capture the dependencies between output variables [36].

One step forecasting, here, is broadly addressed in the first three articles for load demand prediction. Besides, multi-step forecasting with different strategies is discussed in Paper IV for both demand and supply forecasting.

2.1.4.5 Point vs Probabilistic Forecasting

Single-point forecasting results in point outputs; one point at each step. In the other hand, probabilistic forecasting assigns a probability to any of several distinct outputs. The likelihood of predicting is represented by the complete range of probabilities. There are typically three types of probabilistic forecasting known as quantiles, intervals, or density functions. The intervals are usually expressed in two forms: a prediction interval which is related to a prediction, and a confidence interval which is expressed by a parameter. Power utilities used to rely mainly on point load forecasting for their decision-making process. However in recent decades, the application of probabilistic load Forecasting in energy planning and operation is on the increase due to the growth in market competitions and integration of renewable energy sources to the power grid [37].

The focus of this thesis is on point forecasting. Probabilistic forecasting as a wide research topic is recommended as a future research direction.

2.1.4.6 Short-term Load Forecasting Techniques

The forecasting techniques that are employed in different studies for short-term load forecasting of buildings are generally divided into two broad categories: engineering (physical) and data-driven. Engineering models present the thermal performance of the systems and components of the buildings using mathematical equations. EnergyPlus and eQuest simulation software are typical samples of this category. Besides being highly accurate and reliable, they require a high level of details about different parameters of the buildings that are not always available. They also need a high degree of expertise to carry out computations that are costly and elaborate [38]. On the other hand, data-driven approaches do not require such specific knowledge about about the building under study. Instead, they benefit from historical or streaming data. These approaches are further classified into three groups: Statistical, AI-based and ensemble techniques.

Statistical techniques depend on historical data to find a correlation between energy consumption as output and most influential factors as inputs. These methods compared to engineering methods, need a lower level of physical understanding and a smaller number of variables to build the models. Regression techniques, exponential smoothing and time series methods such as Auto Regressive Moving Average (ARMA) and Auto-Regressive Integrated Moving Average (ARIMA) fall in the category of statistical models.

The major drawback of statistical modelling is that its prediction accuracy is dependent on the existence of adequate data samples, and a variety of statistical data assumptions. The sudden changes in load patterns considerably degrade the performance of such techniques. They are also very slow in the scenarios where long-term forecasts with multiple input variables are needed to build the predictive models. Several machine learning techniques have been adopted to address these limitations. The models based on Support Vector Machines (SVM), as well as Classification and Regression Trees (CART), have been successfully applied in energy forecasting applications.

In recent decades, AI-based approaches have been extensively used in load forecasting problems. They rely on both historical and real-time data to build forecasting models. Their main advantage is that they do not need mathematical formulations to manually extract statistical components of load curves. The AI-based models instead, utilize artificial intelligence to capture trends, seasonality and non-linear relationships existent in real load profiles [39]. They, indeed, are created quicker and simpler than physical and statistics models and will offer accurate results if they are trained appropriately [40].

The accuracy of AI models however is limited by the size of training samples. These methods include artificial neural networks, fuzzy logic, expert systems, and optimization-based algorithms such as genetic algorithm and particle swarm optimization.

Artificial neural networks with various specific algorithms, among AI-based techniques, have been extensively applied to load forecasting. The key explanation is their ability to map non-linear relationships that can be found in actual load profiles. Reviews of short- term forecasting using neural network models can be found in [41] and [39]. However, some potential drawbacks of traditional ANNs such as overfitting, sensitivity to random weight initialization and tendency to convergence to local optima [42] led the researchers to investigate on developing more efficient learning algorithms and parameter initialization techniques for the neural networks.

Recently, artificial neural networks with deep architecture have shown improved predictive performance. The deeper networks benefit from additional hidden layers, significantly fewer neurons, improved activation mechanisms, and more efficient learning algorithms [28]. Different versions of deep neural networks such as Conditional Restricted Boltzmann Machine (CRBM) [26], Convolutional Neural Network (CNN) [43] and Long short-term memory network (LSTM) [44] are recently being employed in energy prediction context . A review of deep learning approaches applied to load forecasting is presented in [45].

Moreover, there have been several studies towards the development of hybrid techniques for load forecasting problem. Hybrid approaches aim to overcome the limitations of their incorporating algorithms. Their potential use would be in circumstances where model instability is high and predictive models require sufficient input. There is a variety of hybrid approaches. Some of them combine signal processing and machine learning techniques such as [46] and [47] whereas some create an ensemble of multiple machine learning and/or optimization algorithms such as [38] and [48]. There are also recent applications of ensemble techniques based on deep-learning which are discussed in [49] and [29] for energy prediction problems.

2.2 Renewable Energy Forecasting

2.2.1 Importance of Renewable Energy

Renewable energy refers to clean and useful energy that are collected from renewable resources such as sunlight, wind, waves and geothermal heat. There are several environmental and economic advantages relating to the usage of renewable energy over fossil fuels. First, their supplies are abundant, virtually inexhaustible and recyclable; second, they emit little or low carbon greenhouse gases, thus reducing the risk of global warming, water and air pollution.

In recent years, the adoption of renewable energy applications has increased significantly. There are plenty of cities in the world which are already using renewable resources for transport and industry besides heating and cooling the buildings. According to REN21's 2017 report, over the last ten years the installation and maintenance costs of renewable technologies especially solar PV and onshore wind turbines are falling rapidly (%82 and %39 respectively). It also reports that a growing number of countries across the world are generating more than %20 of their electricity from solar and wind resources. Fig 2.2 shows the share of electricity generation from various types of renewable energy from 2000 to 2018 across the world [50]. The rising trend relating to shares of photovoltaics (PV) and wind turbines indicates the growing popularity and importance of these resources more than before for many countries.

2.2.2 Applications

During the last decades, substantial changes have been made to the conventional electric power grid. Specifically, increasing climate change issues and global warming from fossil fuel power plants have

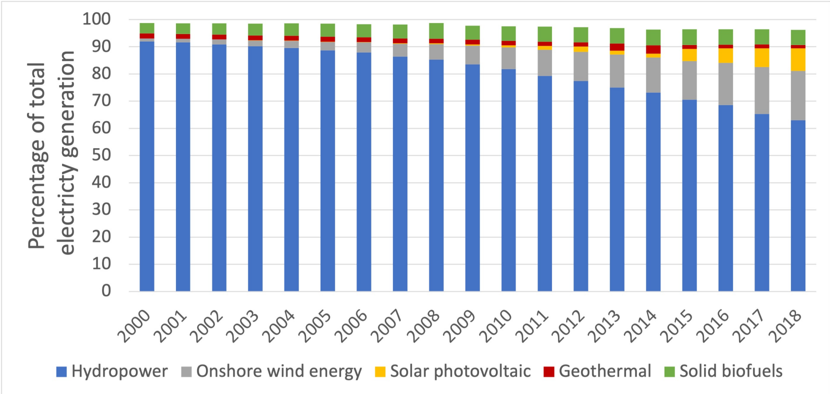

Figure 2.2: The total shares of renewable energy technologies from electricity generation in the world

encouraged the use of renewable energy [51]. Due to numerous advantages of renewable energies, the integration of renewable energy to the power grid is highly expect-able among different power operators. The power generation units from renewable sources have the potential to be distributed among local communities in the power grid. The distributed power generation reduces the dependence of local energy infrastructures from remote sources and a centralized power grid. This would consequently improve the safety and quality of power supply by avoiding weather-related disruptions occurring frequently in the central grid.

At the customer side, the reduced costs in small-scale power generation technologies such as micro solar panels and micro wind turbines besides cost-effective energy trading programs, encourage different customers to use renewable resources to meet their energy needs in more efficient and cost-effective ways.

Although the substitution of renewable energy with fossil fuels has many benefits, the large-scale integration of renewable energy sources raises problems in terms of ensuring the efficiency and sustainability of power systems. Firstly, the load curves of renewable power such as solar and wind energies are highly nonlinear and unpredictable due to their dependencies to volatile weather conditions and local topography. This uncertainty would inevitably increase the reserve capacity of electricity systems, thus making electricity production more expensive. Secondly, the integration of generation units will lower the stability margin of the power system by incorporating more power electronics and accordingly reduction in the rotational inertia of the power systems [52].

Therefore, the forecasting of renewable energy plays an essential role in reducing uncertainties in such situations.

2.2.3 Forecasting Methods

There are several studies in the literature that have examined predictions of solar irradiation, solar power generation, wind speed and wind power. Although the research for solar energy and wind power are evolved separately, they share many forecasting methods at different forecast horizons. The adopted forecasting techniques, similar to the ones in load forecasting problem are generally classified into four categories: physical, statistical, AI-based and hybrid.

At the short-term horizon; from few minutes to few days ahead, physical methods are typically applied for both wind speed [53] and solar irradiation [54] predictions. The physical approaches rely on numerical weather prediction (NWP) models. Their main advantage is that they do not require historical data to provide forecasts. If the technical specifications of the generation unit and NWP are available, the future outputs of the generation unit can be estimated before construction.

The major downside of physical models, however, is the high reliance on NWP, that needs additional information on spatial and temporal resolution. They also suffer from inaccuracy when incorrect data is used as input, requiring them to perform heavy pre-processing tasks. The NWP-based models are further improved through statistical and machine learning-based techniques [55].

Statistical techniques aim to determine statistical relationships between measured observations of renewable energy data within a specific period [56]. Autoregressive Moving Average (ARMA) [57], Autoregressive Integrated Moving Average (ARIMA) [58], [59] and sparse Bayesian [60] are widely investigated in the literature.

The ability of statistical methods, however, is mainly limited to the production of linear models that are not suitable for solving more complicated energy forecasting problems e.g. with longer forecast horizons. AI-based techniques have also been frequently adapted for renewable energy forecasting to overcome the limitations of physical and statistical techniques. Support vector machines [61] and different variants of artificial neural networks [62], [63] have shown successful results in this research area. Recently, deep learning approaches which have achieved high performance in different time-series forecasting tasks, are developed for solar power [64], [65] and wind energy forecasting [66], [67].

The hybrid techniques from the fourth category have also demonstrated promising results by combining the individual methods from the first three categories. In [68] an ensemble model was proposed consisting of data preprocessing and ML algorithms for multistep wind power forecasting. They include Wavelet Packet Decomposition (WPD), Elman Neural Networks (ENN), boosting algorithms and Wavelet Packet Filter (WPF). According to their results, the proposed method outperforms the individual incorporating algorithms. In [69] different forecasting techniques from the CART, linear regression and KNN categories are combined to generate probabilistic solar power forecasts from three solar farms. [70] includes a complete overview of hybrid approaches to solar and wind technology.

This study in Paper IV explores a method for estimating solar PV output of rooftop solar systems. The potential ability to predict other forms of energy data, such as wind power, is also discussed.

Chapter 3

Background

This chapter provides the background of the fundamental concepts used in this thesis. It firstly introduces smart meters which produce a large amount of energy data for performing predictive analytics. Second, it defines machine learning, along with the techniques employed in the thesis for feature selection, load clustering and load forecasting. Third, the concept of load disaggregation and the applied methods to enhance the accuracy of load forecasting are explained. The final section introduces the research technologies which are used to perform our experiments.

3.1 Smart Meters

Smart meters are advanced metering tools which automatically collect electric energy consumption of buildings at frequent intervals e.g. every 10 minutes, 30 minutes, one hour, etc. They provide two-way communications between electric utilities and customers at their premises. The smart meters 'measurements contribute to effective and accurate settlement besides increasing customer knowledge on their energy usage. They create bases of critical information for better monitoring and operation of the power grid. Additionally, some smart meters can report specific events in the grid (e.g. power outages, earth faults) or record certain parameters such as voltage levels, current and power factors).

In the context of load forecasting, the information from smart meters specifically with high resolution (low frequency) will improve insights into granular consumption behaviour of future loads. The detailed analysis of load profiles would lead to more accurate predictions [71].

In all four research papers, we used real smart meter data to develop and train forecasting models. The data sources belong to different households and have been various in terms of observation period and location. For privacy-preserving, the address and ID of electricity customers are anonymized.

3.2 Machine Learning and Deep Learning

Artificial Intelligence (AI) is a sub-discipline of computer science which is combined with engineering to enable a machine to imitate intelligent behaviour of a human being. Artificial intelligence has been broadly applied in developing systems where they mimic goal-oriented human functions like learning, reasoning, understating patterns, etc.

Machine Learning (ML) is a branch of AI which can continuously be modified by learning from data. According to Tom M. Mitchell, as a machine learning pioneer [72], ML is the study of computer algorithms that allow computer programs to build upon themselves through multiple experiences. More precisely, ML algorithms can adapt to new data without human intervention or human assistance.

Furthermore, Deep Learning (DL) refers to a subset of ML which provides more powerful models with larger data sets and more computational tasks. The term 'Deep' in the context of artificial neural networks (ANNs) means the larger number of hidden layers in the structure of the network. The performance of deep models can continuously improve by having access to more data. The deep neural networks unlike shallow networks and traditional ML techniques do not need extensive feature engineering to learn the relationships between inputs and outputs, instead, they automatically learn the features from raw data sequentially or hierarchically. As mentioned in Chapter II, deep-learning-based approaches for energy load predictions are extensively adopted in the literature. The predictive algorithms based on deep neural networks are mainly employed in Paper II [21] and Paper IV.

3.3 Feature Selection Techniques

In machine learning, 'Feature Selection (FS)' refers to a process where a subset of relevant features which contribute most to the prediction, is identified from a set of input data. A feature selection algorithm can be seen as a mechanism involving the search of new feature subsets, and an assessment measure that ranks them.

There are three categories of FS techniques based on the assessment measure. (1) Filter methods that scores a subset of useful features based on a proxy measure such as the Pearson Correlation Coefficient and Mutual Information. Filters are computationally trivial and fast, however, their developed feature set is not customized for any predictive algorithm. (2) Wrapper methods that train a predictive model to rank feature subsets. The subsets producing lower error rates will be given higher scores.

Compared to filters, wrapper methods are more complicated and computationally more intensive, but usually, create an optimal set of features for a particular model or problem. (3) Embedded methods that combine the qualities of filters and wrapper techniques, perform feature selection as part of the predictive model construction. In general, different types of feature selection techniques aim to develop models with higher prediction accuracy, shorter training times, lower variance and a higher level of interpretability [73].

The Feature selection techniques which are investigated in Paper I [20] and Paper IV, are mainly used to define the most influential variables on energy use and generation of households. Furthermore, they were employed to enhance the forecasting performance.

3.4 K-means Clustering

Clustering is a type of unsupervised machine learning algorithm. In an unsupervised approach, the assumptions are derived from samples that do not contain a labelled output variable. Clustering has different kinds of applications including pattern recognition and clustering-based estimation. There are two main categories of clustering analysis: hierarchical and partitional. The former assigns given data samples to the required number of clusters. Hierarchical clustering results in a hierarchically organized series of clusters, which contributes to the final cluster. The latter, in contrast, represents each cluster by a centre which is a descriptive overview of all data points existent within the cluster. Partitional methods split data points into a predefined number of clusters by optimising an objective function. The objective function minimises the distance between the data points and the cluster centre.

K-means clustering [74] is considered as a classic partitional analysis which can manage big data. It is also a simple, flexible and reliable approach for clustering purposes [75]. K-means has investigated in significant research applications related to segmenting customers in the power network. For example, the household profiles are clustered by K-means based on their hourly and daily electricity consummation patterns in [76] and [77] respectively.

In this study, Paper I [20] explains K-means in more details and applies this algorithm to distinguish house profiles based on their daily load variation and daily peak consumption.

3.5 Predictive Techniques

In this part, the forecasting techniques enclosed in the research publications are briefly presented.

3.5.1 Auto-Regressive Moving Average

Auto-Regressive Moving Average (ARIMA) [78] as a time series analysis technique is fitted to a univariate time series data and widely used for predicting future points in the series. ARIMA proposed by Box and Jenkins in 1970, produces a stationary time series by removing trend and seasonality from the original input. In a stationary time series, the statistical properties such as mean, variance and covariance remain constant over time. The forecasts produced by ARIMA are considered as a linear function of the most recent observations and past random errors.

ARIMA compared to regression techniques does not require a set of predictor variables, however, it needs heavy fine-tuning of parameters and usually loses its precision by increasing forecast horizon. To overcome the limitations of ARIMA, the various extensions of ARIMA combined with other forecasting methods have been widely investigated in the context of energy forecasting. Sample applications of the hybrid models with ARIMA adapted to load demand and solar energy prediction are discussed in [79] and [80] respectively. ARIMA was investigated as a baseline time series forecasting technique in Paper IV.

3.5.2 Ridge Regression

Ridge regression [81] belongs to a class of regression techniques which models a linear relationship between multiple predictive variables as input and prediction target as the output. Ridge regression, adds L2 penalty to reduce the complexity of the model when the data is high dimensional or when the correlation between input variables is high. The L2 parameter has the effect of decreasing the coefficient values of certain variables that incorporate least to forecasting. It is calculated using linear least squares to minimize the error [82]. In this study (Paper IV), the predictive ability of Ridge regression as a regularized linear model is compared against the ones of non-linear algorithms in the context of energy forecasting.

3.5.3 Support Vector Regression

The Support Vector Regression (SVR), as a version of Support Vector Machine (SVM) for regression, is widely used for data modelling and time series prediction. This method approximates a function based on observed data to train the model. This linear function can describe the nonlinear relationship between variables in high dimensional feature space. Unlike most traditional forecasting methodologies, there is no model in the strict sense instead, the data drive the prediction. Additionally, SVR minimizes the empirical risk, guarantees the global minimum solution and offers high generalization ability. A mathematical explanation of SVR is provided in [83].

As mentioned in Chapter II, SVR-based models have been successfully applied to both load demand and renewable energy forecasting tasks. The advantages of this technique along with its promising results shown in recent energy-related studies encouraged us to implement this method and compare its performance with those of other proposed techniques in Paper II [21] and Paper IV.

3.5.4 Ensemble Methods

Ensemble methods create meta algorithms by integrating several machine learning algorithms into one powerful forecasting model.

For ensemble methods to be more accurate than any of its members, the base learners have to be as accurate as possible and as diverse as possible.

The technique that integrates predictors is called ensemble learning. Ensemble learning can be performed in different ways:

(1) Bagging (bootstrap aggregation): in this method, several ML algorithms (e.g. Decision Trees) are trained on different random subsets of the data and create the ensemble. To create sub-samples from data, it uses bootstrap sampling that performs sampling with replacement. Therefore, each predictor may be trained on the same training subsets several times. The final estimates of individual learners will be aggregated through 'averaging' for regression and by 'voting' for classification. The resulted meta learner will have less variance compared to the individual predictors. In the context of load forecasting, the bagging method with bootstrap sampling may not be optimal due to inter-dependencies within the historical energy measurements.

(2) Boosting: this method aimed at building a strong learner based on multiple weak learners. The individual models (weak learners) are sequentially trained and fitted on a weighted version of training data. It means the samples which were misclassified or estimated with large errors receive higher weights in the next

iterations. The final output depending on the type of problem would be either weighted majority vote or weighted sum of predictions. The final strong learner will produce lower bias by exploiting the dependencies between the individual learners.

(3) Stacking: Via this technique, at the first level, multiple predictors (either classifiers or regressors) are trained on a subset of training data, then they make predictions on another subset. The produced forecasts, in the next level are further used as the features to train a meta learner. To create a heterogeneous ensemble, the base learners often include various learning algorithms. The meta learner can belong to any category of ML algorithms such as Ridge regression, ANN, Random Forest, etc. The stacked ensemble that learns the optimal weights for combining the first-level preditors would be able to improve accuracy and generalization performance.

In general, the ensembles that produce more accurate results than their members are formed based on the learners with high diversity and accuracy [84]. In our research, Paper IV employs ensemble techniques from all three categories to investigate their limitations and advantages for both load consumption and micro-generation forecasting.

The following lists the applied ensemble algorithms which were evaluated during our research either individually (Paper I [20], Paper II [21], Paper III [22]) or in a combination with other algorithms (Paper IV).

3.5.4.1 Random Forset Regressor (RF)

Random Forest [85] is a commonly used ensemble algorithm belonging to the bagging category. It employs Decision Trees (DTs) [86] as base learners. Each tree is fitted to a sample chosen with replacement from the training set. The trees are further randomized with training on a subset of features rather than all features. The bagging trees will lead to a forest with slightly higher bias but lower variance since the less correlated trees are combined in the forest, thus making the final model more powerful in general.

3.5.4.2 Ada Boost Regressor

The widely used form of boosting approach based on decision trees is called AdaBoost. Ada Boost Regressor, used for regression problems, sequentially train and add multiple one-level decision trees. The process of adding and fitting will continue until either the required number of trees is created or no considerable improvement occurs in terms of training errors. In the end, the output of all estimators in the ensemble are combined by computing 'weighted median'. Ada boost has the potential ability to filter out the features having high predictive capacity. It, therefore, contributes to the reduction of input dimension and improving training efficiency. A comparative study on AdaBoost algorithms applied for times series forecasting is presented in [87].

3.5.4.3 Gradient Boosting Regression Tree

Gradient Boosting Regression Tree (GBRT) [88] is a variant of Tree-based boosting algorithms applied for regression problems. Similar to AdaBoost regressor, GBRT constructs the trees in a stage-wise manner. However, at each step, the decision tree which optimizes a loss function is introduced to the ensemble. The loss function is computed by a gradient descent technique. The output of each new learner is further added to the output of all previously selected trees. Learning from previous mistakes will help the ensemble to produce forecasts with higher accuracy.

3.5.5 Feed Forward Neural Networks

An artificial neural network is a biologically inspired system which consists of a possibly large number of highly interconnected processing elements called artificial neurons. The most common and traditional architecture of the neural networks is the Multilayer Perceptron (MLP) type in whcih neurons are arranged in layers. This architecture is composed of one input layer where the data are introduced to the network, one or more hidden layers where data are processed and one output layer where the results of given input are produced. The neurons in each layer are connected with all the neurons in the previous layer with different weights representing the network knowledge.

In a feed-forward architecture known as FFNN, the outputs of one layer are used as the inputs to the following layer; the information flow is from input to output direction and not vice versa. All neurons except the input neurons use non-linear activation functions to produce the output. A training algorithm is further employed to learn the optimal parameters (weights and bias) of the network.

ANNs typically with the feed-forward structure is reported as successful learning algorithms in forecasting applications for several reasons. First, ANNs are data-driven and self-adaptive methods; they can identify hidden trends within historical observations via a training phase, even though the underlying relation between input and output variables is unclear or hard to describe. Second, after learning the data presented to them, they can correctly generalize it to unseen data, even if the training data contain noisy information. Third, they can numerically approximate any continuous function to the desired accuracy, thus leading them to learn non-linear modelling much better than traditional linear models [12].

All these characteristics suggest ANNs should prove to be particularly useful when dealing with a large amount of volatile load consumption and energy generation data, specifically when we have little prior knowledge about the rules of producing data. [41] and [89] discuss detailed analyses of the use of traditional ANNs in building load forecasting along with the prediction of solar PV and wind power generation. The conventional ANNs with feed-forward and shallow structures (one to two hidden layers) are extensively discussed and evaluated in Paper II [21], Paper III [22] and Paper IV.

3.5.6 Convolutional Neural Networks

A Convolutional Neural Network (CNN) is a subset of deep learning algorithms, primarily applied to image processing problems such as image recognition [90], image classification [91], etc. They have also been useful in predicting energy time-series data [92] and load demand data [93]. CNNs are considered as regularized versions of fully connected networks in a way that they capture high-level patterns in data by identifying and collecting low-level patterns. As compared to other classification algorithms, they need less feature engineering as they can automatically learn the features or main characteristics from the input data.

Similar to a feed-forward neural network, a convolutional neural network for time series data includes an input layer, hidden layers and an output layer. The hidden layers perform one-dimensional convolutions. Through the convolution process, the features are extracted from the input layer through filters, a non-linear transfer function and feature maps. This transformation is typically followed by a pooling operation where the dimension of feature maps is reduced to obtain the important convolution features. Next, a fully connected output layer creates final non-linear combinations of selected features for making estimations by the network.

CNNs are examined and used in Paper III [22] as part of load disaggregation algorithms. Moreover, they are modified as predictive algorithms in Paper IV to provide multi-step energy forecasts. More details on structures of applied CNNs can be found in the given studies.

3.5.7 Recurrent Neural Networks

Following are the list of investigated methods in the category of recurrent neural networks.

3.5.7.1 Long-Short Term Memory Networks

Recurrent Neural Networks (RNNs) are neural networks which use feedback connections among the nodes to remember the values from previous time steps. Thus, this will allow them to learn the temporal aspects of time series data. Each recurrent neuron in a traditional RNN, receives input as well as its output from the previous time step. On long sequences of input, however, as the backpropagation algorithm is used to train these networks; the gradients tend to be exploded or vanished over many time steps. For capturing long-term patterns or independencies in the sequential data, the standard Long-Short Term Memory Network (LSTM) was introduced by Hochreiter and Schmiduber [94] in 1997 and was gradually improved over the years.

LSTM unlike conventional RNN algorithms by using internal memory cells, addresses vanishing or exploding gradient problem by faster convergence and provides a model to store information for long and short periods [95]. LSTM networks have been popular deep algorithms with high precision in the area of sequence learning like natural language translation [96], and speech recognition [97].

Concerning power data displaying apparent features of time series data with cycles, LSTM cell information (long-term data dependencies) can be beneficial for load forecasting. As discussed in Chapter II, LSTM and its variants recently have been successfully adapted for short-term forecasting of residential and commercial loads.

Therefore to address our research problem, we extensively investigate different types of LSTM networks that have demonstrated high precision in time series prediction problems. They include (1) the standard LSTM, investigated in Paper II [21] for one-hour ahead load consumption and daily peak load estimation as well as in Paper IV for the multi-hour ahead load consummation and generation forecasting. The structure of a standard LSTM network is completely explained in the referred papers. (2) Gated Rectified Unit (GRU) and (3) Sequence To Sequence LSTM discussed in Paper IV for multi-hour ahead energy forecasting. The remaining of this section briefly describes the two variants of standard LSTM. More details are provided in the mentioned research study (Paper IV).

3.5.7.2 Gated Recurrent Unit

Gated Recurrent Unit (GRU) is a compact version of LSTM which was proposed by Cho et.al [98] in 2014. It follows the same principles of processing long-term sequences in LSTM, but with fewer gates and parameters. GRU, therefore, compared to LSTM, due to its simplified architecture, can be more efficient in terms of training time. The accuracy of GRU was found to be comparable to that of LSTM on some tasks of speech recognition and natural language processing [99]. GRU-based networks have even exhibited better performance in some smaller data sets [100]. Apart from those listed, they have also recently been used in residential load forecasting [101], and photovoltaic forecasting [102].

3.5.7.3 Sequence To Sequence LSTM

A Sequence-to-Sequence network is a subset of artificial neural networks developed for converting sequences to sequences. These networks which are also known as encoder-decoder networks have been primarily suggested to incorporate machine translation schemes. In such networks, the source language sentences are fed into the encoder while the decoder interprets the destination language sentences. The sequence to sequence network that uses layered recurrent networks was first introduced by I.Sutskever et.al in [96]. This architecture was further adapted for time series forecasting practices, particularly types of forecasting that require multiple steps ahead [103] and [104].

3.6 Load Disaggregation

3.6.1 Applications

Smart energy management systems at residential buildings provide households and grid operators with real-time information about home appliances. The information on operational states and energy consumption of appliances can be beneficial for both sides. Regarding the residential customers, the information would help them diagnose the health of appliances, save energy, reduce their bills and improve their consumption behaviour. The smart grid operators also utilize this information to forecast peak load, develop intelligent control strategies, evaluate appliance energy efficiency and detect faulty devices more effectively [105],[106].

Current household monitoring solutions available in the market, typically follow an intrusive strategy to provide appliance consumption data. They mainly require entering the house to provide monitoring interfaces and install sub-meters for each appliance. In contrast, smart meters, which measure the total consumption of a building, enable non-intrusive load monitoring with minimal maintenance and installation costs. Privacy-preserving of end-users is also another advantage of this method over intrusive solutions.

Nevertheless, smart meters do not provide appliance-level data due to practical constraints. Using energy disaggregation algorithms in conjunction with smart meters data can efficiently provide energy management systems with appliance-level information [107].

Energy disaggregation, also known as Non-Intrusive Load Monitoring (NILM) was originally developed by George G.W.Hart et.al at MIT in the early 1980s. NILM for appliances refers to the process where the power consumption and the operation state of each electric device in a building are estimated from an aggregated power signal. This aggregated signal is often measured with one power meter which monitors all devices.

There have been several studies on NILM approaches to provide appliance-level feedback. Complete surveys on non-intrusive load monitoring techniques for energy disaggregation and energy management are presented in [108] and [109].

Paper III [22] of this thesis, has studied three load disaggregation algorithms to evaluate their capability in improving the household load forecasting. Following are the list of applied techniques. The choice of algorithms was based on their availability in the NILMTK toolkit[1] besides the diversity in their architectures and disaggregation methods.

3.6.2 Applied Techniques

3.6.2.1 Factorial Hidden Markov Model

Hidden Markov Models (HMMs) are a class of probabilistic graphical model that allows us to predict a sequence of unknown (hidden) variables from a set of observed variables. In the realm of load disaggregation, HMMs find the contribution of power of each appliance corresponding to the (hidden) state of the appliance. The states and corresponding appliance power are accessible from sub-meters' measurements. The Hidden Markov model is trained using historical data to estimate the model parameters [110].

[1]NILMTK is a Python-based open-source framework which enables the training and comparison of NILM algorithms across various data sets

Factorial Hidden Markov Model (FHMM) introduced by Michael. J in 1996 is as an extension of HMMs. FHMM models multiple independent hidden state sequences by a probability distribution function. Each appliance in FHMM is described by a Hidden Markov Model containing states and the transitions between them. The average power consumption is determined for each state, and the probability of transitions are calculated. The aggregate power observations are then allocated to devices whose models follow the shape of the appliance signature. The mathematical details are provided in [111].

FHMM's output typically deteriorates with the rise in the number of devices and, subsequently, with the increase in the number of combinations and the amount of computation time.

3.6.2.2 Denoising Auto Encoder

An autoencoder is a specific type of neural network with an unsupervised learning technique. By this technique, typically for dimension reduction, the network discovers efficient data properties (encoding) through ignoring signal 'noise' in the input. The results from encoding phase (code) are further uncompressed into the format which matches best with the original input. The Denoising Autoencoder (DAE) learns how to recover the original input from the partially distorted version.

In the context of load disaggregation, the overall consumption signal is assumed as a noisy representation of the target appliance signal and the DAE excludes the share of the other appliances from the aggregate load signal. More precisely, the DAE receives a window of the mains readings of fixed length and outputs the inferred appliance consumption for the same time window.

The DAE solution to address the energy disaggregation problem is discussed in [112]. Accordingly, the structure of applied DAE in our study was adapted from this reference. The DAE network there consists of one- dimensional CNNs for both encoder and decoder with three dense layers in between.

3.6.2.3 Sequence To Sequence Optimization

A sequence to sequence (seq2seq) architecture, adapted for NILM, converts aggregate energy measurements as input sequence to the target appliance power as the output sequence. The sequences of input and output are framed as sliding windows in such a manner that only a subset of windows can be used for training, thus minimizing the cost of computing. The mapping would enable the network to align the observed trends in the main signal with the appliance signal characteristics while training.

A seq2seq network can employ different types of deep networks including RNN, LSTM and CNNs with various configurations. The architecture of sequence to sequence technique which was used in our evaluation was adapted from a study in [113]. This network employs five one-dimensional CNNs and two fully connected networks with different sizes to extract the target sequence from the input signal. The details are provided in the mentioned study [113].

3.7 Research Technologies

In this section, the programming language and the modern Machine Learning (ML) libraries which are used to implement our solutions are introduced.

3.7.1 Programming Language

'Python', is used in this dissertation as a preferred programming language to perform predictive analytic. Python, developed in 1991, is a popular programming language in the field of machine learning and data science. It is known as a high-level interpreted language which emphasizes readability and less complexity in writing compared to languages such as C++ and Java. Like other modern languages, it benefits from object-orientation, memory management, functional and imperative programming, and strong typing. Its key strengths include open-source implementation, support of multiple programming paradigms and is highly portable across operating systems [2].

3.7.2 ML Libraries

We have used multiple Python libraries for data preprocessing and data analysis. 'Numpy' and 'Scipy' for creating and manipulating arrays and matrices at any size; 'Pandas' as a data analysis library and 'Matplotlib' to plot quality figures in a variety of formats and interactive environments. Moreover, 'scikit-learn' [3] has been employed to perform many of the standard machine learning tasks such as data scaling, clustering, regression and building time-series and conventional ML models such as Regression Trees and SVR algorithms.

To implement different types of artificial neural networks with deep structures, we employed 'Keras' [4] software library. Keras is one of the most powerful and easy-to-use open-source Python libraries which is designed for developing and evaluating deep learning models. Keras was introduced by François Chollet; a Google engineer during a research project known as ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System). Keras mainly acts as an API interface for one of the most prominent and convenient machine learning platforms called TensorFlow [5]. TensorFlow's main features are its multi-layered system of nodes that allows artificial neural networks to be trained flexibly and rapidly on large datasets. It also provides several advanced optimization nodes to search for the parameters that reduce a cost function.

Aside from other aspects, Keras is simple to use and flexible as an extendable Python API, which has motivated us to use it extensively in our experiments.

3.7.3 NILM Library

We have used NILMTK to run the experiments regarding load disaggregation practices (NILM-related techniques). NILMTK as an

[2] https://www.python.org/about/
[3] https://scikit-learn.org/stable/
[4] https://keras.io/about/
[5] https://www.tensorflow.org

open-source platform allows convenient comparisons of the NILM algorithms across different data sets. It comprises of multiple dataset parsers, data set analysis information, algorithms for preprocessing and disaggregation, as well as several assessment indicators. The implementation and configuration of the applied algorithms are thus adapted from NILMTK-contrib repository [6].

[6]https://github.com/nilmtk/nilmtk-contrib

Chapter 4

Contributions

This chapter includes a review of four separate research papers with their respective contributions to the thesis. The description of each paper includes the motivation, methods and results for the task.

4.1 Overview

In this dissertation, we addressed the need for the new frameworks to preserve load balance in smart energy communities. The motivations discussed in each paper present contributions to the dissertation.

Paper I [20] conducts a comparative study on multiple feature selection methods. They support with distinguishing highly relevant factors influencing the consumption behaviour of different households. It uses smart meter data of several houses in Norway to perform the experiments. Paper II [21] uses the most widely used machine learning algorithms in the context of short-term load forecasting to evaluate their prediction and generalization abilities at the household level. A public energy consumption dataset of residential customers in the UK from different eco-social classes was used to train and evaluate the models.

Paper III [22] investigates the applicability of advanced load monitoring techniques (NILM) for improving the accuracy of hourly load forecasting. It performs experiments on two public data sets customized for NILM. Paper IV addresses forecasting of electricity and micro solar power generation at the community level. It describes a framework for the identification of relevant predictive factors for both energy supply and demand. It also studies the development of hybrid forecasting approaches based on the most accurate individual models.

In terms of descriptive and predictive analysis, all papers focus on machine learning techniques for data mining, data analysis and predictive modelling specifically for time-series data. As mentioned in Chapter III, section 3.7, all the experiments have been conducted using Python programming language and advanced Python libraries such as 'Pandas', 'scikit-learn' and 'Keras'. The NILMTK library was also employed to run load disaggregation algorithms on the experimental data sets.

In the remaining of this chapter, we present a summary for each research paper.

4.2 Paper I

Evaluating Feature Selection Methods for Short-term Load Forecasting

This paper was published in Proceedings of 6th IEEE International Conference on Big Data and Smart Computing (BigComp), 2019.

Motivation:

Electricity suppliers need short-term forecasts for their critical decision-making activities such as load flow control, planning supply and scheduling generation. Timely implementation of such decisions ensures the stability of the power network and thus decreases the occurrence of system faults and blackouts. Over the decades, the advancements in mathematical tools and computational power have contributed to the development of forecasting techniques with higher accuracy.

However accurate load forecasting specifically at building level remains a challenging task. Concerning the residential load, the accuracy of forecasting is not only dependent on the forecasting method but also to the several fluctuating factors such as time variables, weather data and consumption habits. In this study, we explore these factors and investigate their diversity among the house profiles with various levels of load volatility. We also study how the relevant features influence on accuracy and performance of a predictive model among different groups of households.

Methodology:

In this study, we followed a two-step approach. In the first step, we collected smart meter data of 23 Norwegian households as a sample dataset and created a set of candidate features based on the literature study. In the second step, we performed load clustering (using K-means algorithm) to distinguish consumption patterns based on daily load volatility. Next, four feature selection techniques were applied to choose a subset of features from the candidate set per house profile. They include 'F-regression', 'Mutual Information (MI)' from filters,' Recursive Feature Elimination(RFE)' from wrappers and 'Elastic Net' from embedded category. To test the effectiveness of FS techniques on improving load forecasting, we evaluated the prediction accuracy of one forecasting algorithm (GBRT) with and without FS results across each house and cluster.

Results:

The identified subset for each house profile consists of both common and non-common factors among the same and different clusters' members. The common factors across the three recognized clusters mostly include load-related features such as 'past load consumption values during previous one and two hours' and 'average usage of past 24 hours' besides some non-load variables such as 'day of the year' and 'outside temperature'. The uncommon features do not belong to a particular category and are found in different categories, e.g., load, time and temperature.

The average of prediction results across clusters' members shows that all FS-techniques improve forecasting performance in terms of accuracy and training time. These results also reveal that no unique

FS technique can contribute to the lowest prediction error for all types of house profiles. However, it outlines the load forecasting of houses with higher load volatility would need more advanced FS techniques such as a combination of MI and RFE.

4.3 Paper II

Short-Term Load Forecasting Using Smart Meter Data: A Generalization Analysis

This paper was published in Processes open access journal, belonging to special issue Clean Energy Conversion Processes, 2020.

Motivation:

There exist different methods addressing the problem of energy prediction in buildings. They are mainly divided into two categories: Engineering and Data-driven (statistical and machine learning). The engineering approaches which represent mathematical models of buildings are highly reliable and accurate but difficult to generalize. The customized model for one building can not apply to another building with different dynamic behaviour. The data-driven techniques (i.e. ML), on the other hand, use consumption data and fit a model from given inputs to desired outputs. The learned rules can be further used to produce predictions for new inputs (e.g. the measurements from the unseen building during training).

In this study, we use smart meter recordings as a source of historical data and evaluate the generalization ability of multiple states of the art ML algorithms in forecasting hourly load and daily peak demand of buildings. They are constructed only on the basis of historical load variables to test the predictive power of these models in the absence of external variables.

Methodology:

To address the problem, four predictive algorithms from various ML categories were chosen. (1) Support Vector Regression and (2) MLP with shallow feed-forward structure (FFNN) from traditional ML. (3) Gradient Boosted Regression Tree (GBRT) from ensemble/CART category and (4) standard LSTM with time steps from the most recent field (deep recurrent artificial neural networks). To train and evaluate the models, we obtained smart meter data of 75 UK houses with hourly resolution over one year period. We also evaluated the sensitivity of the models to various input sizes and the number of lag variables. As the energy data is a type of time series data, we employed a k-split time-series cross-validation technique rather than the standard k-fold method. A comparative study was further performed to assess the predictive and generalization ability of the models across different seasons and within various socio-economic house profiles.

Results:

All the models specifically LSTM and FFNN generalized well; on average they produced low errors in one-hour ahead predictions for unseen houses during training. In terms of estimating daily peak load consumption, the GBRT and LSTM outperformed the others. Regarding the training and tuning times, the GBRT was found as the fastest among the four algorithms. The sensitivity analysis indicates that increasing the number of training houses will boost forecasts as long as the added profiles improve the model's awareness of the test houses. The results also suggest that consumers with lower average annual usage and lower variance in hourly loads produce more stable profiles. An analysis of seasonal forecasts showed that seasons with lower temperature typically come with more load violations, making forecasting for almost all models more challenging.

4.4 Paper III

Improving Load Forecast Accuracy of Households Using Load Disaggregation Techniques

This paper was published in Proceedings of 11.th International Conferences on IEEE Smart Data (SmartData), 2020.

Motivation:

The availability of data with high sampling rates through smart meters has made it possible to produce a comprehensive study of usage patterns and the development of data-driven models with high prediction accuracy. Also, the growing use of smart meter data has allowed non-intrusive load monitoring (NILM) with low installation and maintenance costs. NILM refers to the process where individual load consumption of appliances is determined by measuring the aggregate energy consumption of the house without sub-metering of individual devices. There has been extensive research regarding the development of NILM algorithms however a few have investigated the applicability of NILM in the energy forecasting context. The purpose of this paper is to incorporate NILM outputs (i.e. appliance feedbacks) to construct more reliable and accurate forecasting models.

Methodology:

To reach this goal, we have proposed a hybrid approach consisting of ML and NILM algorithms. In this approach, not only the aggregated energy consumption but also the appliance-level data provided by energy disaggregation algorithms are part of the input to the forecast model. The combinations of two forecasting models (MLP and GBRT) with three states of the art NILM algorithms (Factorial Hidden Markov Model (FHMM), Denoising Auto Encoder (DAE) and Sequence to Sequence optimization) are evaluated in the proposed framework. We chose three houses from two public NILM datasets (UK-DALE and REFIT) to validate the results. The models were trained on one house and tested on both seen and unseen houses during training. There was also a comparative analysis of forecasts with and without disaggregation inputs.

Results:

The combinations of NILM and ML forecasting algorithms have shown promising results in improving forecasting performance. Among the disaggregation methods, the Sequence to Sequence algorithm was more accurate in following the appliance signals for unseen and seen houses. In terms of forecasting techniques, the hybrid models with disaggregation phases; FHMM+MLP and Seq2Seq+MLP, yielded better accuracy in comparison to the models without the disaggregation phase (showing a decrease of at least 80 % and 42 % in MAE, respectively). The findings suggest further study for a wider variety of appliances and recent NILM algorithms based on deep learning.

4.5 Paper IV

An ensemble approach for multi-step ahead energy forecasting of household communities

This paper was accepted for publication in IEEE Access, 2021.

Motivation:

Due to the many advantages of green energy, the use of renewable energy is growing on power networks. On the consumer side, the reduced costs of small-scale power generation units (e.g. solar PV and wind turbine) besides advancements in ICT, would provide opportunities for consumers to meet their energy demands through locally distributed energy sources. However, in such an environment, forecasting energy demand and supply becomes essential to reduce the instability induced by the integration of micro-generation sources. The forecasting task especially with an extended forecast horizon will become challenging due to several factors leading to non-linearity and instability of load demand and supply. In this study, we focus on residential customers and extend the application of ML algorithms to solar PV output forecasting and multi-step ahead prediction.

Methodology:

In this paper, we have proposed a framework to accurately forecast electricity consumption and solar PV generation of household communities at 24 hours ahead. The proposed framework provides a process for designing an effective ensemble of forecast models based on comprehensive analyses of baseline forecasting algorithms. We have used the sequence to sequence LSTM networks as base learners of the ensemble besides the GBRT algorithm as a meta learner. To effectively deal with non-linearities in load consumption and solar energy data, the weather and calendar variables were incorporated into the forecasting process. As a case study, the presented framework was evaluated on energy prediction of residential neighbourhoods in Sydney and Newcastle.

Results:

Through feature selection analysis, the factors like 'Global Horizontal Irradiance (GHI)' and 'air temperature' were found to be highly influential on the estimation of micro-solar power production and load consumption accordingly. Based on the prediction results, the proposed ensemble outperform traditional ensembles and individual best learners for the majority of test communities in terms of both predictive targets. Moreover, with increasing the forecast steps, the ensemble model illustrates more robust predictions. Furthermore, the performance evaluation in different weather conditions and day types show the superiority of the ensemble approaches over single deep neural network models.

4.6 RQ-Findings

This section explains how the research papers have addressed and answered the research questions.

RQ 1: Which factors influence the most on consumption and generation behaviour of prosumers?

To investigate this, we explored the literature study and performed data exploration tasks on real data sets. They include plotting charts and applying feature engineering and feature selection techniques in combination with various predictive models. They are extensively discussed in Section II and Section IV of Paper I and Section III of Paper IV.

RQ 2: What techniques are appropriate and highly accurate

for prediction of energy consumption and small-scale generation of households?

The related techniques along with benefits and drawbacks are mainly discussed in papers II and IV. Paper II analyses and compares the prediction performance of different ML techniques for household short-term load forecasting. Paper IV tests various baselines to discover the most useful and most reliable algorithms for prediction at low aggregation levels.

RQ 3: How can we improve the performance of successful predictive models at building/local levels?

To explore this, we have proposed new frameworks in Paper II and Paper IV, Section II, Part C. Paper II discusses a hybrid approach to improve hourly load forecasting at the building level. It solves the issue by incorporating load disaggregation results to load forecasting process. Paper IV presents an advanced ensemble approach based on recurrent deep neural networks to boost the forecasting of the community load.

Chapter 5

Conclusion and Future Work

This chapter outlines the conclusion of the research study as well as future research directions.

5.1 Conclusion

This thesis proposes an extension to research work to analyse energy profiles and improve the predictive models at smart energy communities through predictive analytic. Two parts of the thesis are addressed with three research questions in four articles.

It studies the importance of load and renewable energy forecasting in current and future power grids to establish the research objectives in real-world scenarios. It has made it possible to perform practical and diverse predictive analytics using real data sets obtained from smart meters of residential buildings in different countries. A variety of data analysis tasks such as data cleaning, feature engineering, feature selection and clustering is performed to build predictive models with high accuracy and generalization ability.

In Paper I [20], we have analyzed residential demand profiles in the clusters with various load volatility levels. Besides, the highly relevant factors for electricity usage across the established clusters have been identified. The results suggest that the types of influential factors may slightly vary based on the consumption behaviour of residents. However, there are shared variables such as recent load observations and air temperature that are recognized as highly informative inputs for most of ML algorithms. In Paper IV, the impact of lagged load and weather variables has been also shown to be effective in solar power forecasting.

Paper II [21] adapts the most influential features from previous work (Paper I) and uses them as the inputs to various ML algorithms customized for time series forecasting. We also evaluated the generalization capability of ML techniques in real scenarios where one or a group of trained models can be used for load forecasting of houses whose profiles are not seen by the models during training. The low prediction errors in daily peak consumption yielded by LSTM network and the GBRT along with their high generalization ability would suggest deployment of such techniques for energy management systems at local levels. Being uni-variate also make these models appropriate for the situations where access to external information is limited or when the model development is costly e.g. for online learning.

Paper III [22] investigates a hybrid solution to improve short-term load forecasting of households. It focuses on using load disaggregation techniques to provide conventional forecasting models with appliance-level feed-backs. The potential application of the proposed technique could be in scenarios where employing deep forecasting models is not efficient or cost-effective. It may occur due to lack of access to long-term historical data or the necessity of fast training in time-sensitive environments.

Paper IV extends the application of LSTM networks and its variants for multi-step ahead and multi-target forecasting. It creates an ensemble-based model to benefit from the capabilities of both deep neural networks and regression tree algorithms. The experimental results show the advantages of the proposed approach for accurate prediction of both electricity consumption and micro-energy generation of aggregate loads at the community level. Forecast information can be essential for load balancing and solar energy trading in smart microgrids.

5.2 Future Work

A possible extension of this work can be transferring centralized data analytics structure to a decentralized one. This transformation will offer many benefits, including online learning for energy forecasting. Online learning can be applicable in time-sensitive environments such as the network edge at IoT devices where the computation resources i.e. time, memory and processing power are limited. The forecasting methods based on streaming data will be more adaptive to sudden changes in the consumption and generation patterns of prosumers thus, being more efficient in preserving stability in smart energy communities.

Other research direction in future would be investigating probabilistic forecasting methods for energy load forecasting. Probabilistic load forecasting by assigning probabilities to future load values raises knowledge on volatility and unpredictability of potential loads. Therefore, it can be significantly advantageous at low aggregation levels where power demand and supply fluctuate more and exhibit high degrees of uncertainty. This can be direct extensions to the work presented in Paper I [20], Paper II [21] and Paper IV.

Further development on load disaggregation techniques is suggested as the future study in Paper III [22]. It would be useful to adopt NILM techniques based on advanced deep learning to improve prediction accuracy. Besides, applying transfer learning for existing NILM methods allows them to be effectively used for new domains such as a new appliance or a new test house from another country. The NILM techniques with high generalization ability could thus add more flexibility and efficiency into a forecasting framework.

References

[1] *PAT RESEARCH*. URL: https://www.predictiveanalyticstoday.com/what-is-predictive-analytics/#google_vignette (visited on 12/01/2020).

[2] Damir Novosel Michael I. Henderson and Mariesa L. Crow. *Electric Power Grid Modernization, Trends, Challenges and Opportunities*. URL: https://www.cmu.edu/epp/irle/readings/henderson-novosel-crow-electric-power-grid-modernization.pdf (visited on 11/01/2017).

[3] Stephen Khan. *Transmission Loss*. URL: https://theconversation.com/we-calculated-emissions-due-to-electricity-loss-on-the-power-grid-globally-its-a-lot-128296 (visited on 12/01/2020).

[4] United States Environmental Protection Agency. *Energy and the Environment*. URL: https://www.epa.gov/energy/centralized-generation-electricity-and-its-impacts-environment (visited on 03/13/2018).

[5] Guobin Xu, Wei Yu, David Griffith, Nada Golmie, and Paul Moulema. "Toward integrating distributed energy resources and storage devices in smart grid." In: *IEEE internet of things journal* 4.1 (2016), pp. 192–204.

[6] AJ Dinusha Rathnayaka, Vidyasagar M Potdar, Tharam Dillon, and Samitha Kuruppu. "Framework to manage multiple goals in community-based energy sharing network in smart grid." In: *International Journal of Electrical Power and Energy Systems* 73 (2015), pp. 615–624.

[7] Verena Heinisch, Mikael Odenberger, Lisa Göransson, and Filip Johnsson. "Organizing prosumers into electricity trading communities: Costs to attain electricity transfer limitations and self-sufficiency goals." In: *International Journal of Energy Research* 43.13 (2019), pp. 7021–7039.

[8] Eunice Espe, Vidyasagar Potdar, and Elizabeth Chang. "Prosumer communities and relationships in smart grids: A literature review, evolution and future directions." In: *Energies* 11.10 (2018), p. 2528.

[9] Lulu Wen, Kaile Zhou, Shanlin Yang, and Xinhui Lu. "Optimal load dispatch of community microgrid with deep learning based solar power and load forecasting." In: *Energy* 171 (2019), pp. 1053–1065.

[10] V Lavanya and N Senthil Kumar. "A review: control strategies for power quality improvement in microgrid." In: *Int J Renew Energy Res* 8 (2018), pp. 150–65.

[11] Luis Hernandez, Carlos Baladron, Javier M Aguiar, Belén Carro, Antonio J Sanchez-Esguevillas, Jaime Lloret, and Joaquim Massana. "A survey on electric power demand forecasting: future trends in smart grids, microgrids and smart buildings." In: *IEEE Communications Surveys & Tutorials* 16.3 (2014), pp. 1460–1495.

[12] G Peter Zhang. "Time series forecasting using a hybrid ARIMA and neural network model." In: *Neurocomputing* 50 (2003), pp. 159–175.

[13] Filipe Rodrigues, Carlos Cardeira, and Calado. "The Daily and Hourly Energy Consumption and Load Forecasting Using ANN Method: A Case Study Using a Set of 93 Households in Portugal." In: *Energy Procedia* 62 (2014), pp. 220–229.

[14] Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Wil L Kling. "Comparison of machine learning methods for estimating energy consumption in buildings." In: *2014 international conference on probabilistic methods applied to power systems (PMAPS)*. IEEE. 2014, pp. 1–6.

[15] Aoife M Foley, Paul G Leahy, Antonino Marvuglia, and Eamon J McKeogh. "Current methods and advances in forecasting of wind power generation." In: *Renewable Energy* 37.1 (2012), pp. 1–8.

[16] Cristian-Dragos Dumitru and Adrian Gligor. "Daily Average Wind Energy Forecasting Using Artificial Neural Networks." In: *Procedia Engineering* 181 (2017), pp. 829–836. ISSN: 1877-7058. DOI: `https://doi.org/10.1016/j.proeng.2017.02.474`.

[17] Sthitapragyan Mohanty, Prashanta K. Patra, Sudhansu S. Sahoo, and Asit Mohanty. "Forecasting of solar energy with application for a growing economy like India: Survey and implication." In: *Renewable and Sustainable Energy Reviews* 78 (2017), pp. 539–553. ISSN: 1364-0321. DOI: `https://doi.org/10.1016/j.rser.2017.04.107`. URL: `http://www.sciencedirect.com/science/article/pii/S1364032117305853`.

[18] Abinet Tesfaye Eseye, Jianhua Zhang, and Dehua Zheng. "Short-term photovoltaic solar power forecasting using a hybrid Wavelet-PSO-SVM model based on SCADA and Meteorological information." In: *Renewable Energy* 118 (2018), pp. 357–367. ISSN: 0960-1481. DOI: `https://doi.org/10.1016/j.renene.2017.11.011`. URL: `http://www.sciencedirect.com/science/article/pii/S0960148117311126`.

[19] Cyril Voyant, Ted Soubdhan, Philippe Lauret, Mathieu David, and Marc Muselli. "Statistical parameters as a means to a priori assess the accuracy of solar forecasting models." In: *Energy* 90 (2015), pp. 671–679.

[20] A. M. Pirbazari, A. Chakravorty, and C. Rong. "Evaluating Feature Selection Methods for Short-Term Load Forecasting." In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2019, pp. 1–8. DOI: `10.1109/BIGCOMP.2019.8679188`.

[21] Aida Mehdipour Pirbazari, Mina Farmanbar, Antorweep Chakravorty, and Chunming Rong. "Short-Term Load Forecasting Using Smart Meter Data: A Generalization Analysis." In: *Processes* 8.4 (2020). ISSN: 2227-9717. URL: `https://www.mdpi.com/2227-9717/8/4/484`.

[22] A. M. Pirbazari, M. Farmanbar, A. Chakravorty, and C. Rong. "Improving Load Forecast Accuracy of Households Using Load Disaggregation Techniques." In: *2020 International*

Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics). 2020, pp. 843–851. DOI: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00140.

[23] Tao Hong et al. "Energy forecasting: Past, present, and future." In: *Foresight: The International Journal of Applied Forecasting* 32 (2014), pp. 43–48.

[24] Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. "Neural networks for short-term load forecasting: A review and evaluation." In: *IEEE Transactions on power systems* 16.1 (2001), pp. 44–55.

[25] L. Ghods and M. Kalantar. "Methods for long-term electric load demand forecasting; a comprehensive investigation." In: *2008 IEEE International Conference on Industrial Technology*. 2008, pp. 1–4. DOI: 10.1109/ICIT.2008.4608469.

[26] Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Wil L Kling. "Deep learning for estimating building energy consumption." In: *Sustainable Energy, Grids and Networks* 6 (2016), pp. 91–99.

[27] Chun-Nam Yu, Piotr Mirowski, and Tin Kam Ho. "A sparse coding approach to household electricity demand forecasting in smart grids." In: *IEEE Transactions on Smart Grid* 8.2 (2016), pp. 738–748.

[28] Rodrigo F Berriel, Andre Teixeira Lopes, Alexandre Rodrigues, Flavio Miguel Varejao, and Thiago Oliveira-Santos. "Monthly energy consumption forecast: A deep learning approach." In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 4283–4290.

[29] Xueheng Qiu, Ye Ren, Ponnuthurai Nagaratnam Suganthan, and Gehan AJ Amaratunga. "Empirical mode decomposition based ensemble deep learning for load demand time series

forecasting." In: *Applied Soft Computing* 54 (2017), pp. 246–255.

[30] Vasudev Dehalwar, Akhtar Kalam, Mohan Lal Kolhe, and Aladin Zayegh. "Electricity load forecasting for Urban area using weather forecast information." In: *2016 IEEE International Conference on Power and Renewable Energy (ICPRE)*. IEEE. 2016, pp. 355–359.

[31] Magnus Dahl, Adam Brun, Oliver S Kirsebom, and Gorm B Andresen. "Improving short-term heat load forecasts with calendar and holiday data." In: *Energies* 11.7 (2018), p. 1678.

[32] Franklin L Quilumba, Wei-Jen Lee, Heng Huang, David Y Wang, and Robert L Szabados. "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities." In: *IEEE Transactions on Smart Grid* 6.2 (2014), pp. 911–918.

[33] Min Han and Y Wang. "Analysis and modeling of multivariate chaotic time series based on neural network." In: *Expert Systems with Applications* 36.2 (2009), pp. 1280–1290.

[34] George Atsalakis and Katerina Tsakalaki. "Simulating annealing and neural networks for chaotic time series forecasting." In: *Chaotic Modeling and Simulation* 1 (2012), pp. 81–90.

[35] Matteo Sangiorgio and Fabio Dercole. "Robustness of LSTM neural networks for multi-step forecasting of chaotic time series." In: *Chaos, Solitons & Fractals* 139 (2020), p. 110045.

[36] Gianluca Bontempi, Souhaib Ben Taieb, and Yann-Aël Le Borgne. "Machine Learning Strategies for Time Series Forecasting." In: *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures*. Ed. by Marie-Aude Aufaure and Esteban Zimányi. Springer Berlin Heidelberg, 2013, pp. 62–77.

[37] Tao Hong and Shu Fan. "Probabilistic electric load forecasting: A tutorial review." In: *International Journal of Forecasting* 32.3 (2016), pp. 914–938.

[38] Zeyu Wang, Yueren Wang, and Ravi S Srinivasan. "A novel ensemble learning approach to support building energy use prediction." In: *Energy and Buildings* 159 (2018), pp. 109–122.

[39] Arjun Baliyan, Kumar Gaurav, and Sudhansu Kumar Mishra. "A review of short term load forecasting using artificial neural network models." In: *Procedia Computer Science* 48 (2015), pp. 121–125.

[40] Hamid R Khosravani, Marıa Del Mar Castilla, Manuel Berenguel, Antonio E Ruano, and Pedro M Ferreira. "A comparison of energy consumption prediction models based on neural networks of a bioclimatic building." In: *Energies* 9.1 (2016), p. 57.

[41] Ahmad S Ahmad, Mohammad Y Hassan, Md P Abdullah, Hasimah A Rahman, F Hussin, Hayati Abdullah, and Rahman Saidur. "A review on applications of ANN and SVM for building electrical energy consumption forecasting." In: *Renewable and Sustainable Energy Reviews* 33 (2014), pp. 102–109.

[42] Fateme Fahiman, Sarah M Erfani, Sutharshan Rajasegarar, Marimuthu Palaniswami, and Christopher Leckie. "Improving load forecasting based on deep learning and K-shape clustering." In: *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2017, pp. 4134–4141.

[43] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. "Short-term residential load forecasting based on LSTM recurrent neural network." In: *IEEE Transactions on Smart Grid* 10.1 (2017), pp. 841–851.

[44] Mengmeng Cai, Manisa Pipattanasomporn, and Saifur Rahman. "Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques." In: *Applied Energy* 236 (2019), pp. 1078–1088.

[45] Abdulaziz Almalaq and George Edwards. "A review of deep learning methods applied on load forecasting." In: *2017 16th IEEE international conference on machine learning and applications (ICMLA)*. IEEE. 2017, pp. 511–516.

[46] Jinliang Zhang, Yi-Ming Wei, Dezhi Li, Zhongfu Tan, and Jianhua Zhou. "Short term electricity load forecasting using a hybrid model." In: *Energy* 158 (2018), pp. 774–781.

[47] Yanbing Lin, Hongyuan Luo, Deyun Wang, Haixiang Guo, and Kejun Zhu. "An ensemble model based on machine learning methods and data preprocessing for short-term electric load forecasting." In: *Energies* 10.8 (2017), p. 1186.

[48] Wei Tang Zhi Zhou Wu. "Ensembling neural networks, Many could be better than all." In: *Artificial Intelligence* 137 (2002), pp. 239–263.

[49] Xueheng Qiu, Le Zhang, Ye Ren, Ponnuthurai N Suganthan, and Gehan Amaratunga. "Ensemble deep learning for regression and time series forecasting." In: *Computational Intelligence in Ensemble Learning (CIEL), 2014 IEEE Symposium on*. IEEE, 2014, pp. 1–6.

[50] *IRENA*. URL: `https://www.irena.org/Statistics/View-Data-by-Topic/Capacity-and-Generation/Technologies` (visited on 12/01/2020).

[51] H. Wang, J. Ruan, G. Wang, B. Zhou, Y. Liu, X. Fu, and J. Peng. "Deep Learning-Based Interval State Estimation of AC Smart Grids Against Sparse Cyber Attacks." In: *IEEE Transactions on Industrial Informatics* 14.11 (2018), pp. 4766–4778. DOI: `10.1109/TII.2018.2804669`.

[52] Huaizhi Wang, Zhenxing Lei, Xian Zhang, Bin Zhou, and Jianchun Peng. "A review of deep learning for renewable energy forecasting." In: *Energy Conversion and Management* 198 (2019), p. 111799.

[53] Sultan Al-Yahyai, Yassine Charabi, and Adel Gastli. "Review of the use of numerical weather prediction (NWP) models for wind energy assessment." In: *Renewable and Sustainable Energy Reviews* 14.9 (2010), pp. 3192–3198.

[54] David P Larson, Lukas Nonnenmacher, and Carlos FM Coimbra. "Day-ahead forecasting of solar power output from photovoltaic plants in the American Southwest." In: *Renewable Energy* 91 (2016), pp. 11–20.

[55] Bri-Mathias Hodge, Carlo Brancucci Martinez-Anido, Qin Wang, Erol Chartan, Anthony Florita, and Juha Kiviluoma. "The combined value of wind and solar power forecasting improvements and electricity storage." In: *Applied Energy* 214 (2018), pp. 1–15.

[56] Stefano Ferrari, Massimo Lazzaroni, Vincenzo Piuri, Loredana Cristaldi, and Marco Faifer. "Statistical models approach for solar radiation prediction." In: *2013 IEEE international instrumentation and measurement technology conference (I2MTC)*. IEEE. 2013, pp. 1734–1739.

[57] Rui Huang, Tiana Huang, Rajit Gadh, and Na Li. "Solar generation prediction using the ARMA model in a laboratory-level micro-grid." In: *2012 IEEE third international conference on smart grid communications (SmartGridComm)*. IEEE. 2012, pp. 528–533.

[58] SN Singh, Abheejeet Mohapatra, et al. "Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting." In: *Renewable energy* 136 (2019), pp. 758–768.

[59] Ilhami Colak, Mehmet Yesilbudak, Naci Genc, and Ramazan Bayindir. "Multi-period prediction of solar radiation using ARMA and ARIMA models." In: *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE. 2015, pp. 1045–1049.

[60] Yun Wang, Haibo Wang, Dipti Srinivasan, and Qinghua Hu. "Robust functional regression for wind speed forecasting based on Sparse Bayesian learning." In: *Renewable Energy* 132 (2019), pp. 43–60.

[61] Jianwu Zeng and Wei Qiao. "Short-term solar power prediction using a support vector machine." In: *Renewable Energy* 52 (2013), pp. 118–127.

[62] Jianzhou Wang, Na Zhang, and Haiyan Lu. "A novel system based on neural networks with linear combination framework for wind speed forecasting." In: *Energy Conversion and Management* 181 (2019), pp. 425–442.

[63] Amir Sharifian, M Jabbari Ghadi, Sahand Ghavidel, Li Li, and Jiangfeng Zhang. "A new method based on Type-2 fuzzy neural network for accurate wind power forecasting under uncertain data." In: *Renewable energy* 120 (2018), pp. 220–230.

[64] Woonghee Lee, Keonwoo Kim, Junsep Park, Jinhee Kim, and Younghoon Kim. "Forecasting solar power using long-short term memory and convolutional neural networks." In: *IEEE Access* 6 (2018), pp. 73068–73080.

[65] André Gensler, Janosch Henze, Bernhard Sick, and Nils Raabe. "Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks." In: *2016 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE. 2016, pp. 002858–002865.

[66] Chun-Yang Zhang, CL Philip Chen, Min Gan, and Long Chen. "Predictive deep Boltzmann machine for multiperiod wind speed forecasting." In: *IEEE Transactions on Sustainable Energy* 6.4 (2015), pp. 1416–1425.

[67] Huaizhi Wang, Haiyan Yi, Jianchun Peng, Guibin Wang, Yitao Liu, Hui Jiang, and Wenxin Liu. "Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network." In: *Energy conversion and management* 153 (2017), pp. 409–422.

[68] Yanfei Li, Huipeng Shi, Fengze Han, Zhu Duan, and Hui Liu. "Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy." In: *Renewable energy* 135 (2019), pp. 540–553.

[69] Azhar Ahmed Mohammed, Waheeb Yaqub, and Zeyar Aung. "Probabilistic forecasting of solar power: An ensemble learning approach." In: *International Conference on Intelligent Decision Technologies*. Springer. 2017, pp. 449–458.

[70] Ye Ren, PN Suganthan, and N Srikanth. "Ensemble methods for wind and solar power forecasting—A state-of-the-art review." In: *Renewable and Sustainable Energy Reviews* 50 (2015), pp. 82–91.

[71] Antorweep Chakravorty, Chunming Rong, Pål Evensen, and Tomasz Wiktor Wlodarczyk. "A distributed gaussian-means clustering algorithm for forecasting domestic energy usage." In: *2014 International Conference on Smart Computing*. IEEE. 2014, pp. 229–236.

[72] Tom M Mitchell. "Artificial neural networks." In: *Machine learning* 45 (1997), pp. 81–127.

[73] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection." In: *Journal of machine learning research* 3.Mar (2003), pp. 1157–1182.

[74] Sanghamitra Bandyopadhyay and Sriparna Saha. *Unsupervised classification: similarity measures, classical and metaheuristic approaches, and applications*. Springer Science & Business Media, 2012.

[75] Ali Al-Wakeel, Jianzhong Wu, and Nick Jenkins. "K-means based load estimation of domestic smart meter measurements." In: *Applied energy* 194 (2017), pp. 333–342.

[76] Fintan McLoughlin, Aidan Duffy, and Michael Conlon. "A clustering approach to domestic electricity load profile characterisation using smart metering data." In: *Applied energy* 141 (2015), pp. 190–199.

[77] Joshua D Rhodes, Wesley J Cole, Charles R Upshaw, Thomas F Edgar, and Michael E Webber. "Clustering analysis of residential electricity demand profiles." In: *Applied Energy* 135 (2014), pp. 461–471.

[78] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford university press, 2012.

[79] Hongzhan Nie, Guohui Liu, Xiaoman Liu, and Yong Wang. "Hybrid of ARIMA and SVMs for short-term load forecasting." In: *Energy Procedia* 16 (2012), pp. 1455–1460.

[80] Sharif Atique, Subrina Noureen, Vishwajit Roy, Vinitha Subburaj, Stephen Bayne, and Joshua Macfie. "Forecasting of total daily solar energy generation using ARIMA: A case study." In: *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)*. IEEE. 2019, pp. 0114–0119.

[81] Arthur E Hoerl and Robert W Kennard. "Ridge regression: Biased estimation for nonorthogonal problems." In: *Technometrics* 12.1 (1970), pp. 55–67.

[82] AK Md Ehsanes Saleh, Mohammad Arashi, and BM Golam Kibria. *Theory of ridge regression estimation with applications*. Vol. 285. John Wiley & Sons, 2019.

[83] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[84] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[85] Leo Breiman. "Random forests." In: *Machine learning* 45.1 (2001), pp. 5–32.

[86] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[87] Devon K Barrow and Sven F Crone. "A comparison of AdaBoost algorithms for time series forecast combination." In: *International Journal of Forecasting* 32.4 (2016), pp. 1103–1119.

[88] Jerome H Friedman. "Greedy function approximation: a gradient boosting machine." In: *Annals of statistics* (2001), pp. 1189–1232.

[89] Jesús Ferrero Bermejo, Juan F Gomez Fernandez, Fernando Olivencia Polo, and Adolfo Crespo Márquez. "A review of the use of artificial neural network models for energy and reliability prediction. A study of the solar PV, hydraulic and wind energy sources." In: *Applied Sciences* 9.9 (2019), p. 1844.

[90] Zheng Yi. "Evaluation and implementation of convolutional neural networks in image recognition." In: *Journal of Physics: Conference Series*. Vol. 1087. 6. IOP Publishing. 2018, p. 062018.

[91] Mahbub Hussain, Jordan J Bird, and Diego R Faria. "A study on cnn transfer learning for image classification." In: *UK Workshop on computational Intelligence*. Springer. 2018, pp. 191–202.

[92] Irena Koprinska, Dengsong Wu, and Zheng Wang. "Convolutional neural networks for energy time series forecasting." In: *2018 international joint conference on neural networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[93] Kasun Amarasinghe, Daniel L Marino, and Milos Manic. "Deep neural networks for energy load forecasting." In: *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*. IEEE. 2017, pp. 1483–1488.

[94] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[95] Daniel L Marino, Kasun Amarasinghe, and Milos Manic. "Building energy load forecasting using deep neural networks." In: *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2016, pp. 7046–7051.

[96] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks." In: *arXiv preprint arXiv:1409.3215* (2014).

[97] Alex Graves and Navdeep Jaitly. "Towards end-to-end speech recognition with recurrent neural networks." In: *International conference on machine learning*. PMLR. 2014, pp. 1764–1772.

[98] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the properties of neural machine translation: Encoder-decoder approaches." In: *arXiv preprint arXiv:1409.1259* (2014).

[99] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. “Empirical evaluation of gated recurrent neural networks on sequence modeling.” In: *arXiv preprint arXiv:1412.3555* (2014).

[100] Mirco Ravanelli, Philemon Brakel, Maurizio Omologo, and Yoshua Bengio. “Light gated recurrent units for speech recognition.” In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 2.2 (2018), pp. 92–102.

[101] Muhammad Sajjad, Zulfiqar Ahmad Khan, Amin Ullah, Tanveer Hussain, Waseem Ullah, Mi Young Lee, and Sung Wook Baik. “A novel CNN-GRU-based hybrid approach for short-term residential load forecasting.” In: *IEEE Access* 8 (2020), pp. 143759–143768.

[102] Yusen Wang, Wenlong Liao, and Yuqing Chang. “Gated recurrent unit network-based short-term photovoltaic forecasting.” In: *Energies* 11.8 (2018), p. 2163.

[103] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. “LSTM-based encoder-decoder for multi-sensor anomaly detection.” In: *arXiv preprint arXiv:1607.00148* (2016).

[104] Shengdong Du, Tianrui Li, Yan Yang, and Shi-Jinn Horng. “Multivariate time series forecasting via attention-based encoder–decoder framework.” In: *Neurocomputing* 388 (2020), pp. 269–279.

[105] Wan He and Ying Chai. “An empirical study on energy disaggregation via deep learning.” In: *2016 2nd International Conference on Artificial Intelligence and Industrial Engineering (AIIE 2016)*. Atlantis Press. 2016.

[106] Bin Zhou, Wentao Li, K.W. Chan, Yijia Cao, Yonghong Kuang, Xi Liu, and Xiong Wang. “Smart home energy management systems: Concept, configurations, and scheduling strategies.” In: *Renewable and Sustainable Energy Reviews* 61 (Aug. 2016), pp. 30–40.

[107] Felan Carlo C Garcia, Christine May C Creayla, and Erees Queen B Macabebe. "Development of an intelligent system for smart home energy disaggregation using stacked denoising autoencoders." In: *Procedia Computer Science* 105 (2017), pp. 248–255.

[108] Anthony Faustine, Nerey Henry Mvungi, Shubi Kaijage, and Kisangiri Michael. "A survey on non-intrusive load monitoring methodies and techniques for energy disaggregation problem." In: *arXiv preprint arXiv:1703.00785* (2017).

[109] I Abubakar, SN Khalid, MW Mustafa, Hussain Shareef, and M Mustapha. "Application of load monitoring in appliances' energy management–A review." In: *Renewable and Sustainable Energy Reviews* 67 (2017), pp. 235–245.

[110] G. A. Raiker, S. B. Reddy, L. Umanand, A. Yadav, and M. M. Shaikh. "Approach to Non-Intrusive Load Monitoring using Factorial Hidden Markov Model." In: *2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)*. 2018, pp. 381–386. DOI: 10.1109/ICIINFS.2018.8721436.

[111] Ahmed Zoha, Alexander Gluhak, Michele Nati, and Muhammad Ali Imran. "Low-power appliance monitoring using factorial hidden markov models." In: *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE. 2013, pp. 527–532.

[112] Jack Kelly and William Knottenbelt. "Neural nilm: Deep neural networks applied to energy disaggregation." In: *Proceedings of the 2nd ACM international conference on embedded systems for energy-efficient built environments*. 2015, pp. 55–64.

[113] Chaoyun Zhang, Mingjun Zhong, Zongzuo Wang, Nigel Goddard, and Charles Sutton. "Sequence-to-point learning with neural networks for non-intrusive load monitoring." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

Paper I: Evaluating Feature Selection Methods for Short-Term Load Forecasting

A. M. Pirbazari, A. Chakravorty and C. Rong, "Evaluating Feature Selection Methods for Short-Term Load Forecasting," *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, 2019, pp. 1-8, doi: 10.1109/BIGCOMP.2019.8679188.

This paper is not included in Brage due to copyright restrictions.

Paper II:
Short-Term Load Forecasting Using Smart Meter Data: A Generalization Analysis

Short-Term Load Forecasting Using Smart Meter Data: A Generalization Analysis

4036 Stavanger, Norway

A.M. Pirbazari[1], M. Farmanbar[1], A. Chakravorty[1], C. Rong[1]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger

Abstract:

Short-term load forecasting ensures the efficient operation of power systems besides affording continuous power supply for energy consumers. Smart meters that are capable of providing detailed information on buildings energy consumption, open several doors of opportunity to short-term load forecasting at the individual building level. In the current paper, four machine learning methods have been employed to forecast the daily peak and hourly energy consumption of domestic buildings. The utilized models depend merely on buildings historical energy consumption and are evaluated on the profiles that were not previously trained on. It is evident that developing data-driven models lacking external information such as weather and building data are of great importance under the situations that the access to such information is limited or the computational procedures are costly. Moreover, the performance evaluation of the models on separated house profiles determines their generalization ability for unseen consumption profiles. The conducted experiments on the smart meter data of several UK houses demonstrated that if the models are fed with sufficient historical data, they can be generalized to a satisfactory level and produce quite accurate results even if they only use past consumption values as the predictor variables. Furthermore, among the four applied models, the ones based on deep learning and ensemble techniques, display better performance in predicting daily peak load consumption than those of others.

1 Introduction

Over the last decade, smart meters have been rapidly deployed around the world. Around 86 million and 11 million smart meters have been installed by large and small suppliers in the US and UK respectively by the end of 2018 [1], [2]. Almost 90% of these meters were installed for residential customers. One of the main objectives of smart metering in residential sectors is to encourage users to consume less energy by raising awareness about their consumption levels. Smart meters provide information on cost and amount of energy consumption in near real-time for both suppliers and consumers.

Regarding the households, huge amounts of fine-grained data on the use of energy not only provide them with more accurate bills but also with valuable information on their electricity consumption habits and time-based pricing rates. This information through demand response and incentivization programs would help them to reduce their energy usage during peak hours and schedule their appliances according to electricity prices. High-resolution data generated by smart meters, on the other hand, provide suppliers with several controlling functions such as power quality monitoring and power loss identification. Moreover, it opens many doors of opportunities in electricity load analysis such as load forecasting with high accuracy at lower aggregation levels [3], [4].

Electrical load forecasting is the prediction of the load demand that an electricity consumer will have in the future. Load forecasts help suppliers to balance supply and demand and to ensure the reliability of power grids at the time of power deficiency. Load forecasts are also important for electricity traders to balance their electricity purchase and sales [5].

Load forecasting is performed in a wide range of time horizons aiming at different targets: short-term load forecast (a few minutes to 1 day ahead) to adjust supply and demand; medium-term load forecast (1 day to 1 year ahead) to plan outage and maintenance and long-term load forecast (more than 1 year ahead) to plan the development of power infrastructures. Load forecasting is also performed in various aggregation levels when it is applied to the areas with different geographical scales such as a country, city, small communities or a building. The forecasting task becomes more challenging when it comes to lower aggregation levels such as a building level since, many fluctuating factors affect a building's energy consumption with varying degrees such as weather parameters, building properties, Heating, Ventilating and Air-Conditioning (HAVC) facilities and the consumption behavior of occupants [6], [7].

A large number of studies on accurate short-term load forecasting has been reported in recent years due to its impact on the efficient operation of power systems and the economy. Furthermore, many studies have benefited from smart metering data to develop more advanced models for load forecasting at individual building levels. The methods for predicting building energy consumption generally are classified into two categories: engineering (physical) and data-driven techniques. Engineering methods use mathematical equations to present the physical components and thermal performance of buildings. However, they need high details about different parameters of the buildings that are not always available. Moreover, a high level of expertise is required to perform expensive and elaborate computations.

On the other hand, data-driven approaches do not need such detailed data about the simulated building and instead learns from real-time or historical data. These approaches are further classified into two groups: statistical and AI-based techniques [8], [9]. Statistical methods use historical data as an aim for correlating energy consumption with the most important variables as inputs. Therefore, a larger amount of historical data with high quality plays an important role in the effectiveness of statistical models. Traditional linear statistical models such as Gaussian mixture models (GMM), Conditional demand analysis (CDA), Regression models and auto autoregressive moving average (ARMA) and ARIMA, have remained the baseline for time series prediction with widespread use in many applications [10].

Although it is easy to use statistical techniques, the basic assumption of such models is based on the fact that time series are considered linear and therefore follow a specifically known distribution of statistics. Numerous machine learning models have been developed to overcome these limitations. The models based on Support Vector Machines (SVM), as well as Classification and Regression Trees (CART), are among the successful machine learning techniques used in time series forecasting and energy applications.

Over the past decades, many researchers have investigated the application of AI-based techniques in forecasting problems. Among AI-based techniques, Artificial Neural Networks (ANNs) with different structures have been widely applied in the load forecasting domain [11]. ANNs similar to statistical methods use historical data to build a model. However, with hidden layer structures and learning ability offer several advantages over statistical and classical machine learning techniques for time series forecasting. They are considered data-driven and self- adaptive methods which can capture subtle and functional patterns through a training process on historical records of data, even if the underlying relationship between input and output variables is complex or unknown. Nevertheless, the neural networks with shallow structures have the disadvantage of assuming that all inputs and outputs are independent of each other, even when dealing with sequential data [12].

Recent studies in time series forecasting have shown the better performance of prediction models using neural networks with deep architecture. Long Short-Term Memory (LSTM) network which was first proposed by [13], is a variation of deep learning concept which was designed specifically to learn the short-term and long-term dependencies present in sequential data. LSTM has been popular with excellent accuracy in the realm of sequence learning like natural language translation and speech recognition [14]. In recent years there has been an increase in the number of studies on the application of LSTM networks and their variants in short-term load forecasting.

According to the literature, most classical and AI-based methods specifically deep techniques developed for load forecasting, require sufficient historical load data for training. At the building level, they mainly have access to the historical energy consumption of the building under study and utilize it for training the model. Subsequently, for performance evaluation, they use the future profile of the same house or building.

Similarly, in higher aggregation levels such as a community or a substation level, the models are trained on aggregated historical consumption of buildings and are tested on the future profile of the same buildings. Typically, the low testing error of the models guarantees the precise prediction and the small difference between train and test error ensures the models' generalization ability. Moreover, many of the studies use additional information and build multivariate models based on consumer behavior [15], weather and calendar parameters [16] appliance measurements [17], etc. to improve the forecasting accuracy.

However, there are still some issues about the forecasting accuracy and generalization ability of such models which have not been largely addressed. For example, to what extent the generalization ability can be expanded or trusted and what happens to the model forecasting accuracy if we only provide them with consumption data.

The first question focuses on the scalability of the models; how successful the forecasting models are when tested on a different profile that they are not previously trained on. The test profiles could be quite different from the trained ones in terms of consumption patterns or average daily and yearly consumption. This may happen in scenarios when historical information on a building's energy consumption is not accessible and we can still rely on predictive models trained on available historical profiles. For instance, if a new house profile is added to a community, or a new smart meter has been installed. The model developed in these situations can also be less expensive in terms of complexity and computation time. The second question focuses on how powerful a model can be if we only have access to anonymized data on historical energy consumption due to privacy issues or lack of other data sources.

This paper investigates the mentioned scenarios with a focus on short-term load forecasting at an individual building level. For this purpose, we develop four baseline models to predict hourly residential load demand and evaluates their predictive accuracy and generalization ability in the given scenarios. The models are chosen from the category of most-widely used machine learning methods for energy prediction known as ANNs, Support vector Machines, regression trees (CART) and LSTM with standard architecture. They are trained on consumption data of various load profiles and tested on unseen houses with different levels of hourly load volatility.

Furthermore, the sensitivity of the models on the size of training data and the number of input variables will be tested and a comprehensive analysis of forecasting results will be provided. The developed models are expected to learn various load profiles relying on the built-in information in time series data and are aimed at improving generalization ability and increasing model robustness. The models that produce low-prediction errors on multiple houses can further be used as representative predictive models for a group of houses in a community. In demand response applications, this can potentially remove the need to build separate forecasting models per house profile within the community of houses.

The paper is structured as follows. Section 2 provides an overview of the literature. Sections 3 and 4 briefly presents the architecture and design considerations of the implemented forecasting techniques. Then, Section 5 introduces the performance metrics for model evaluations. The dataset used in our analysis is described in Section 6. The experimental results and discussion are provided in Section 7. The paper ends with a conclusion in Section 8.

2 Related Work

There are many forecasting models that have been investigated and proposed since the 1970's for energy predation. Among them, statistical techniques have been extensively applied in load demand forecasting problems. For example, in [18] the authors developed one-day-ahead forecasts on hourly and daily electricity loads of a house using both simple and multiple regression analyses. They utilized weather parameters as the predictor variables and concluded that models trained on the daily dataset provide more accurate forecasting results. S.Sp. Pappas et al. in [19], proposed a method for electricity consumption and price forecasting using AutoRegressive Moving Average (ARMA) models based on adaptive multi-model partitioning theory. Their results show that the proposed method could apply to online modeling and noisy input data.

There have been some hybrid load forecasting approaches that are based on statistical techniques. For instance, XiaoshuLu et al. [20]

presented a hybrid model based on a physical–statistical approach to improve forecast accuracy in energy and building applications. The physical model was developed to define the physical concepts of energy streams while the statistical technique was designed to consider model inconsistencies and specific diversity of buildings.

Support Vector Regression has also been applied in time series prediction as well as power load demand forecasting. In [21] three Support Regression Models and an improved SVR variant utilizing optimization algorithms were used to predict the day-ahead electricity load. The models' effectiveness centered on the small size of training data and their online learning functionality. In [22] we can find a comprehensive overview of SVR applications in time series prediction as well as power load demand forecasting.

Many researchers have also attempted to apply the Classification and Regression Trees (CART) techniques to improve the load forecast accuracy. For instance, Lahouar.A. et al. [23] proposed a model based on random forests for short term load forecast with special attention to load profile, holidays and customer behavior. Similarly, researchers in [7], [24] utilized environmental and calendar features to develop a method for electric load forecasting based on Decision Tree and algorithms.

Recent studies have shown the better performance of prediction models using AI-based techniques due to their ability to learn non-linearities between inputs and outputs. Among AI-based techniques, artificial neural networks have been successfully applied in the forecasting domain. Nasreen K. Ahmed et al. [25] performed an empirical comparison of machine learning models for time series forecasting. In addition to the classical techniques, they analyzed several variations of artificial neural networks. The experiments demonstrated that multilayer perceptron and the gaussian process regression outperform other state-of-the-art models.

There are also some studies that discuss the hybridization of different ANN approaches and are successfully applied to short-term load forecasting. In [26] Hamid R. Khosravani et al. developed hybrid models based on different neural network architectures and genetic algorithms with several optimization parameters to predict electric power demand at the Solar Energy Research Center. The comparison results with an autoregressive baseline model reveal that the models based on the multi-objective genetic algorithm outperformed the model based on computational and empirical methods with lower complicity. Kuihe Yang et al. [27] proposed an ANN-based method with fuzzy logic to develop models with fewer complexities and to improve the accuracy of forecasts.

There have been numerous studies that utilized optimization algorithms to optimize the structural and training parameters of ANNs in forecasting problems. For example, Chaturvedi et al. [28] have demonstrated the effectiveness of training neural networks with a Genetic algorithm as an optimization strategy. A review study on different variants of artificial neural networks in the field of short-term load forecasting emphasizes that a combination of neural networks with evolutionary algorithms could outperform the singles models in terms of forecasting accuracy [29].

Over the last decade, neural networks with deep structure have increasingly attracted the attention of researches in prediction problems. Compared to shallow networks, they benefit from many hidden layers, exponentially fewer neurons, better activation functions, and parameter initialization techniques as well as effective learning algorithms. Different versions of deep neural networks are recently being employed in energy prediction context, especially LSTM networks and their variance due to their capability to capture the temporal behavior of time series data. Daniel. M et al. [30] investigated the effectiveness of LSTM-based architectures for building level energy load forecasting. They applied two standard and Sequence to Sequence (S2S) architectures for the hourly forecast of a residential load dataset with one-minute and one-hour resolutions. Experimental results showed that the standard LSTM performing better in one-hour resolution data while S2S performed well on both datasets.

In another study by Kong et al. [31] on short-term residential load forecasting an LSTM-based framework has been assessed for both individual and aggregated prediction levels. The comparison results with several state-of-the-work approaches demonstrated the superiority of LSTM for individual residential load forecasting. In terms of aggregated (substation) level, the aggregation of all individual forecasts yielded better results than the direct forecast of aggregated loads. Agrawal et al. [32] Introduced a deep-structure RNN-LSTM network at a higher aggregation level; ISO New England energy market using daily, monthly and weekly features to produce hourly predictions over a one-year period.

Similar to the approaches using shallow ANNs, some studies explored the combination of LSTM with other models or optimization algorithms. For instance, in [33] a CNN-LSTM neural network was proposed to predict the energy consumption of residential buildings with higher accuracy. The CNN layer was used to extract complex features influencing energy consumption and the LSTM layer was fed with the features to model the temporal information in time series components. Mamun et al. [34] and Bouktif et al. [35] investigated the effectiveness of hybrid deep neural networks based on LSTM and genetic algorithms for load forecasting on the energy market and metropolitan's electricity consumption data sets. Application of feature selection in [35] proved that using only optimal lagged features as the input to the LSTM model produces the lowest forecasting error for both medium-term and long-term horizons.

3 Modelling Techniques

In this paper, four modeling techniques are used for energy load forecasting: Support Vector Regression (SVR) with Radial Basis Function kernel, Gradient Boosting Regression Trees (GBRT) driven from Classification and Regression Tree (CART) analysis, feedforward neural networks (FFNNs) and LSTM networks. The first two methods belong to the category of classical machine learning techniques and the other two belong to AI-based machine learning techniques with shallow and deep structures respectively. In the following, the detailed information about each model is provided.

3.1 Support Vector Regression (SVR)

SVR is an extension of the support vector machine (SVM) algorithm for numeric prediction or regression tasks. SVM is one of the popular machine learning algorithms used for classification tasks. The SVR identifies and optimizes the generalization bounds given for regression [36]. The formulation of SVR for time series prediction is expressed as follows. Given training data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ where x_i are input vectors and y_i are the associated output value of x_i, then the SVR is an optimization problem as follows:

$$\min \frac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}(\epsilon_i + \epsilon_i^\star),$$

$$Subject\ to\ y_i - (\omega^T\Phi(x_i) + b) \leq \epsilon + \zeta_i,$$
$$(\omega^T\Phi(x_i) + b) - y_i \leq \epsilon + \epsilon_i^\star,$$
$$\zeta_i, \epsilon_i^\star \geq 0, i = 1, \ldots, l$$

Where xi maps to a higher dimensional space and ζ_i is the upper training error ($\epsilon_i^\star$ is the lower) subject to the ϵ-insensitive tube $|\omega^T\Phi(x_i) + b| \leq \epsilon$. The parameters that control the output of the regression are the error cost C, the width of tube ϵ, and the mapping function, Φ [37]. The constraints imply that most data xi are put in the tube $|y_i - (\omega^T\Phi(x_i) + b)| \leq \epsilon$. If xi is not in the tube, there is an error ζ_i or $\epsilon_i^\star$ which we would like to minimize the objective function. ϵ is always zero in traditional least-square regression and data is not mapped to higher dimensional spaces. The SVR formulation theory is similar concerning SVM, and once equipped, the SVR will produce predictions using the following formula:

$$f(x) = \sum_{i=1} l\Theta_i\Phi(x, x_i) + b$$

We used a Radial Basis Function (RBF) as the kernel function due to its ability to capture non-linear relationships between inputs and outputs. The RBF kernel on two samples x and x', represented as feature vectors in the input space, is defined as:

$$K(x, x') = exp(\frac{-\|x - x'\|^2}{2\sigma^2}),$$

Where $\|x - x'\|^2$ is the squared Euclidean distance between the two feature vectors and σ is a free parameter.

3.2 Gradient Boosted Regression Tree

Gradient Boosted Regression Trees (GBRT) is a powerful data-driven technique based on a constructive ensemble strategy and is widely used in non-parametric prediction problems. The GBRT algorithm is a variant of Gradient Boosting Machine (GBM) for regression trees which was originally derived by Friedman (2002) [38].

Two main algorithms define the GBRT model: the decision tree models as the base (weak) learners and gradient boosting algorithm to consecutively fit new models aiming at reaching to more accurate estimation [39].

The target of GBM algorithm is to find an approximation $F\hat{(x)}$ to a function $F(x)$ that minimizes a loss function (y, p); where y is the real output, and p is the target value. The loss function that is selected in our problem, is the squared error $L2$ function as the commonly used loss functions for continuous targets and expressed as follows:

$$l(y,p) = \frac{1}{2}(y_i, F(x_i))^2$$

The negative gradient is simply computed as follows:

$$-[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}] = y_i - F(x_i)$$

The simplicity of the gradient computation will facilitate the residual refitting of the GBM algorithm. The concept behind this loss function is to put penalties on large residuals while neglecting small deviations from the target outputs.

In the GBM algorithm with decision trees as the base learners, the first step is to construct a base tree $h(x; a)$ using a training dataset $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ with size N. Then for the iterations from 1 to M, the negative gradients are computed and a new tree $h(x; a_m)$ is fitted. Each tree is further updated according to the best gradient step p_m and is added to the ensemble. After M iterations, all the regression trees which added sequentially to the ensemble form the output of the algorithm as a combination of weak learners [40].

3.3 Feed Forward Neural Network (FFNN)

An ANN is a system of processing units (neurons) that can be linked together in different ways and estimate various non-linear and arbitrary patterns. In a feed-forward architecture (FFNN), there is no feedback and intra-layer connections between neurons. The weights and bias of the network are estimated using a training algorithm such as the back-propagation algorithm. This algorithm measures the error of output every time and feeds back this information to the network to reduce the error up to an acceptable predefined value. Further, more details on back-propagation algorithms are described in [41].

The input values in an MLP structure are weighed through weight matrices and the output of neurons is determined through an activation function. The structure of the Artificial Neural network that we used in our study illustrated in Figure 1. In the Feed Forward Neural Network illustrated above, given an input sequence $x = (x_1, ..., x_T)$ indicating consumption values from previous T time steps, it computes output y at the next time step y_{T+1} by the following equations:

$$y_i = f_1[\sum_{i=1}^{T} x_i w_i] + b_i,$$

$$y_{T+1} = f_2[\sum_{i=1}^{n} y_j w_j] + b_j$$

Where w_i denotes the input to hidden weight vector, w_j denotes hidden to output weight vector, f_1 refers to a non-linear hidden activation function while f_2 refers to a linear function. b_i and b_j denote bias vectors, and n is the number of neurons in the hidden layer.

3.4 Long-Short Term Memory Network (LSTM)

The LSTM is a variant of Recurrent Neural Network (RNN) which is specially designed for time series data. The RNNs are neural networks that use feedback connections among the nodes to remember the values from previous time steps. Therefore, they will be able to

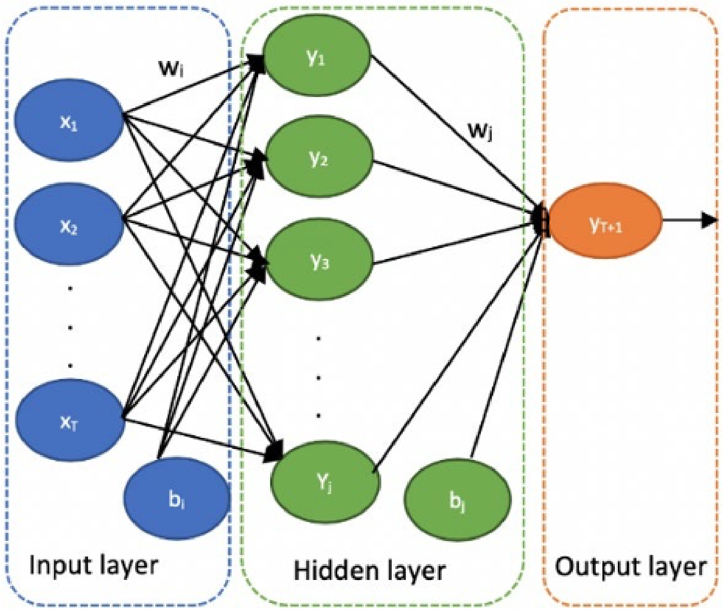

Figure 1: Architecture of applied Artificial Neural Network (ANN)

capture the temporal behavior of time series data. Each neuron, in a conventional RNN, receives the input and its output from the previous step However, on long sequences, they have the problems of vanishing or exploding of gradients over many time steps. The LSTM addresses this problem and empowers RNNs algorithms using internal memory cells [17], [42]. They converge faster and utilize memory cells to store information for long and short periods of time. Regarding power data showing obvious characteristics of time series data with cycles, the history information from LSTM can be advantageous to load forecasting. The structure of the LSTM Network applied to our problem is illustrated in Figure 2.

In an LSTM network, given an input sequence $= (x_1, ..., x_T)$, it computes an output as follows:

$$y_{t+1} = W_h y h_t + b_y$$

where $W_h y$ denotes the hidden-output weight matrix, b_y denotes bias vector and h_t denotes the hidden vector and is computed from the

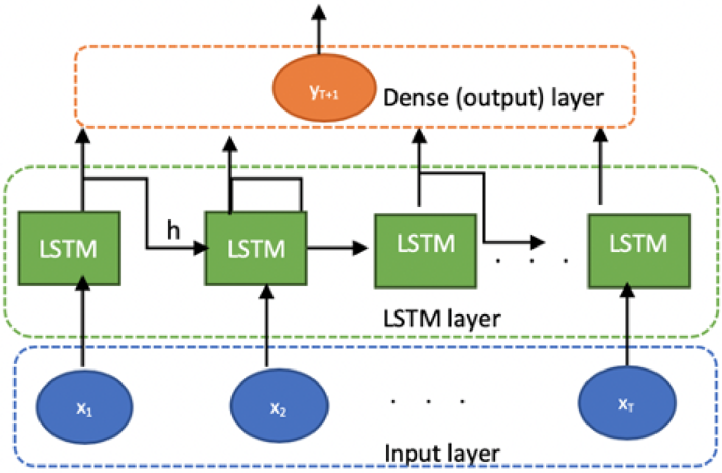

Figure 2: Architecture of applied Long Short-Term Memory (LSTM)

LSTM cell (block). A common LSTM block illustrated in Figure 3. An LSTM cell has three gates: an input gate to identify important information and preserve it in a long-term memory called the cell state C_t, an forget gate to decide what information needs to be forgotten from the previous cell sate C_{t-1} and an output gate to decide what to send to the next sequence.

Once an input x_t enters the LSTM cell, it is passed through a logistic sigmoid function and input gate i_t [42]:

$$i_t = \sigma(W_i[\,c_{t-1}, h_{t-1}, x_t]\, + b_i)$$

Then the output of forget gate is computed as:

$$f_t = \sigma(W_f[\,c_{t-1}, h_{t-1}, x_t]\, + b_f)$$

To scale the output of LSTM activation function, the output gate o_t is expressed as:

$$o_t = \sigma(W_o[\,c_t, h_{t-1}, x_t]\, + b_o)$$

The transient 'memory' value of the activation function, c_t is given as:

$$c_t = i_t \otimes d_t + f_t \otimes d_{t-1})$$

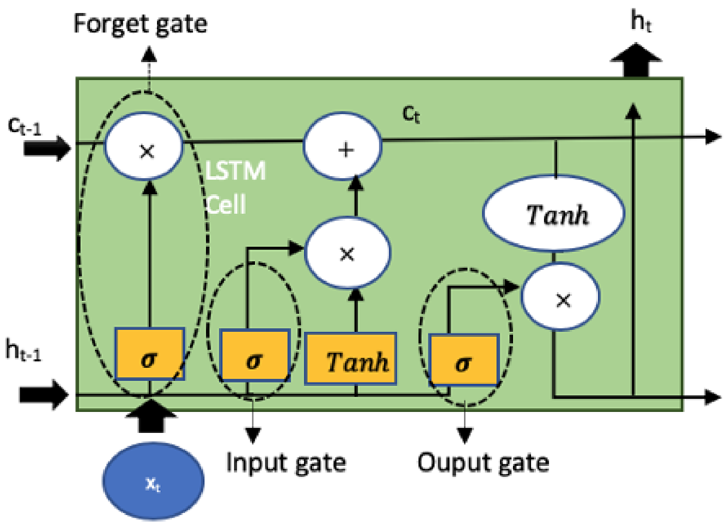

Figure 3: Architecture of an LSTM cell

Here d_t in the input vector of input gate and computed as:

$$d_t = \sigma(W_d[c_{t-1}, h_{t-1}, x_t] + b_d)$$

Where $\otimes$ denotes the element-vise multiplication of the vectors. The LSTM output $h_t$at time step t finally is computed as:

$$h_t = o_t \otimes \tanh c_t.$$

During the training process, the weight matrices W_i, W_f, W_o and W_d and bias vectors b_i, b_f, b_o and b_d are learned by an optimization algorithm.

4 Data Normalization and Parameter Tuning

As mentioned in Section 1, the purpose of this research is to implement the models which are independent of external factors. Therefore, we feed all the models with the past load consumption values in the previous time steps known as load lags. Although all lag variables (features) have the same scale, we scaled data for the FFNN and LSTM networks using Minmax normalization, however, for the other two algorithms (SVR and GBRT) we used the original data. The main reason is that in the AI-based networks normalizing or standardizing the input data usually prevents computational problems and improves the functionality of training algorithms. The scaling method transforms the value of each variable between zero and one as follows:

$$\hat{y} = \frac{y - y_{\min}}{y_{\max} - y_{\min}}$$

For ANN and LSTM as AI-based networks, this transformation is highly recommended to prevent computational problems and improve the functionality of training algorithms. While the choice of scaling for the other two models depends on the problem and the scale of features. After testing the performance of SVR and GBRT on the validation set, with and without data normalization we found that SVR performs better without scaling while GBRT was not significantly affected, with and without data normalization. Therefore, we decided to use the scaled data for all models except for the SVR. The forecasts were rescaled to the initial scale after the prediction process.

For the SVR, with RBF kernel function we tuned the parameters of C, epsilon, and, gamma through a grid search approach. We considered four candidate values for each parameter; 11,050,100 for C, 0.001, 0.01, 0.1, 0.2 for gamma and 0.1, 0.2, 0.3, 0.4 for epsilon. In total, for each given scenario in the experiments, we tested $4 \times 4 \times 4 = 64$ SVR models on the validation set.

For the GBRT model, one of the important parameters that need to be regularized is the number of weak learners (trees). The development of a model with a large number of weak learners would lead to lower regression error, but higher complexity and risk of overfitting. Furthermore, the speed of learning which scales the contribution of each learner is another influential parameter that needs to be set to reduce complexity and computation time. The lower learning rate would normally require fewer learners, thus making the ensemble model simpler with higher generalization ability [40]. To determine the optimum number of trees and the learning rate we employed a grid search strategy to compare the generalization error under each combination of these parameters. According to that, for each variant of the GBRT model, the number of trees was set among the range of 150,200 and 300 and the learning rate was chosen among a range between 0.04 and 0.1.

The tree depth is also another parameter that requires tuning to avoid the overfitting problem. The maximum number of variables for decision splits and the minimum number of records for leaf nodes are the characterizing parameters for defining the tree depth. Generally, the tree depth has a high impact on the overfitting problem, when decision trees are trained on a few observations with a large number of attributes. For our problem, the training set has a large number of records and a few numbers of features (reduces the chance of overfitting. Therefore, we set the maximum number of features for the best split to the input size and set the leaf size as the default value of 'one' considered in the SkLearn library. Nevertheless, to compute the optimal value for the maximum depth of each tree we again used a grid search algorithm and set the candidate values in a range of 2 to 4.

Regarding the FFNN we considered a single hidden layer with n input nodes corresponding to the number of input (lag) variables. To reduce the computation time, we did not tune the number of nodes in the hidden layer, instead, we considered it as twice as the number of input nodes plus one as discussed and suggested in [43]. However, we tuned two parameters related to the training process by grid search: the optimization algorithm and weight initialization technique. We selected 'Adam', 'NAdam' and 'RMSprop' as the candidate values of optimization algorithm along with 'Uniform', 'normal', 'Golrot Normal' as the candidates for weight initialization technique. The linear activation function is used in the Dense layer and The Rectified Linear Units (ReLU) [44] function is used in the hidden layer. A batch size of 128 training samples and a number of 70 epochs (iterations) were chosen for the learning process of each variant. Finally, a learning ratio of 0.001 was set in each iteration for the convergence.

For the LSTM, we did not tune various hyperparameters because of the high computational costs. However, the number of LSTM units as one of the most important hyperparameters related to the network structure was tuned using the validation dataset. The optimal value of this parameter for each LSTM variant was chosen among the candidate values of 5, 10, 15 and 20. The number of features was set to one and the number of timesteps was chosen as the number of lag variables. Adaptive Moment Estimation (ADAM) function was used for optimization due to its computational efficiency and its ability to optimize models with a large number of parameters [45]. The linear activation function is used in the Dense layer before the output of all units and the ReLU activation function was used for the recurrent step. ReLU function is monotonic and half rectified, which assigns zero to any negative values. This has the advantage of not generating vanishing or exploding gradients. However, it can cause dead neurons; therefore, we used the dropout layer between LSTM and Dense (output) layer with the rate of 0.2 to reduce the negative effect of dead neurons which may hurt the training phase. Since LSTM has stronger learning ability than a shallow neural network, higher batch size of 256 and fewer number of epochs (50) was set for training the network.

5 Error Metrics

To evaluate the performance of a forecasting technique, forecasting error is calculated. The lower the forecasting error, the higher the performance of the model. The forecasting error is the difference between the actual observation and the predicted value. There are many error metrics that are proposed for calculating the forecasting error and comparing the performance of time-series forecasting techniques. In this study, we used three such metrics—Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Average Scaled Error (MASE).

If $\hat{y}_t$ is the prediction value and y_t is the actual value at time t and n is the number of test observations, we can define the three metrics as the following:

$$MAE(t) = \frac{1}{n}\sum_{t=1}^{n} |y_t - \hat{y}_t|$$

$$RMSE(t) = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}}$$

$$MASE = \frac{1}{N}\sum_{t=1}^{n}\left[\frac{|y_t - \hat{y}_t|}{\frac{1}{n-1}\sum_{i=2}^{n}|y_i - y_{i-1}|}\right]$$

The MAE calculates the magnitude of the errors on average and ignores whether the prediction values are higher or lower than the real values. Thus, MAE gives equal importance (weight) to all individual differences. The RMSE on the other hand, penalizes large errors by calculating the squared error before averaging them. The MASE was proposed by [46] is introduced as a more applicable error metric and as an alternative to some metrics like Mean Absolute Percentage Error (MAPE) when the observation or prediction values are zero. The MAPE is commonly used as a loss function in model evaluation because it can interpret the relative error. However, the problem with MAPE can occur when there

are zero values in the series and there will be a division by zero. For such sequences, MASE is appropriate as it never produces infinite or unknown values. In this alternative, each actual value of the series in the MAPE formula can be replaced by the average of all actual values of that series.

In addition to these metrics, we added another error metric to our evaluation which is particularly defined for demand forecasting problems, applied in [47] Daily Peak Mean Average Percentage Error (DpMAPE). The DpMAPE measures how accurate is the model in forecasting daily peak consumption. The information about peak time and peak consumption values is highly important for energy management systems for saving grid costs through peak shaving services. The DpMAPE computes the relative difference (percentage) between the daily peak consumption and predicted daily peak value expressed by the following equation:

$$DpMAPE = |\frac{y_{\max} - \hat{y}_{\max}}{y_{\max}}| \times 100\%$$

Finally, since each forecasting model is tested under different scenarios and produces different values for the defined measures, we need to have a combined metric to assess the best variant of each model. This metric is also adapted from [47] and calculates a cumulative weighted error (CWE) based on four defined metrics as follows:

$$CWE = \frac{(RMSE + MAE + MASE + DpMAPE/100)}{4}$$

The CWE is further used to compare the prediction performance of the best variants among different predictive models.

6 Smart Metering Data and Statistical Analysis

The introduced models are evaluated on a subset of energy consumption data set for short-term load forecasting. The original dataset is collected from smart meters installed in 5567 households in London, that took part in the UK Power Networks led Low Carbon London project between November 2011 and February 2014 [48]. The participants in the project were chosen as a representative sample of the greater London population.

The dataset includes recordings from 110 blocks of houses containing energy consumption (in kWh) with the frequency of half-hour, unique household identifier, as well as date and time. The blocks are grouped into 18 categories known as the ACORN (acorn) groups. The social factors and population behavior of each type and category provide precise and valuable information about the households in the given category. A comprehensive and detailed report on the ACORN classification can be found in [49]. For this study, we have chosen seven blocks belonging to five acorn categories known as A, B, C, D and E.

According to the definitions in [49], consumers in groups A, B and C are referred to as 'Affluent achievers'; they live in big houses located in the wealthy and suburban region. Group C is called 'Mature money' and belongs to the retired couples who live in rural towns and villages mainly in detached or semidetached houses. On the other hand, households in groups D and E which are called 'Rising prosperity', are not that wealthy, but younger, educated and living in major cities.

In the preprocessing step, we discarded the houses with a large number of missing records and unusual information. To be precise, among the 347 house profiles existing in the seven blocks, we primarily chose 220 buildings with missing records fewer than a week. From the remaining, the houses with zero mean consumption, indicating no consumption and those with zero standard deviation implying flat consumption were filtered out. Furthermore, the house profiles with unusual total annual consumptions, over 20,000 KWh and less than 2000 KWh as well as total daily consumption of fewer than 3 KWh for more than a month, were discarded.

Finally, from the 180 remaining houses, we randomly selected 15 houses from 5 acorn groups, and therefore a total of 75 house profiles were picked for further study. In the final preprocessing step, linear interpolation was performed on the house profiles containing small gaps from 1 to 24 h. For each building, the energy reading for the year 2013, due to the fewer number of missing records was chosen. Accordingly, the number of observations in each dataset turned to 356 days 24 h = 8760 and the total number of observations in all 75 houses turned to $75 \times 8760 = 657,000$.

Figure 4 illustrates the energy readings of fifteen sample houses in the dataset belonging to different acorn groups over one-year. As we can see they demonstrate different amounts of hourly consumption (ranging between zero and five KW/h), as well as various consumption patterns over the same year (2013).

It is obvious that the short-term load forecasting models are aimed at predicting accurate peak load or energy consumption. However, one of the main influential factors in short-term load forecasting at the household level is the load volatility which simply means deviations from the average consumption. These deviations arise frequently in a residential house because its energy consumption is usually influenced by various factors such as temperature, utilized appliances and consumption habits.

In the context of load forecasting, higher load violation increases the complexity of the load profile, thereby making an accurate load forecasting more complicated. Load analysis of the existing profiles in terms of load volatility will assess in advance which house profiles are potentially more challenging to forecast.

Figures 5 and 6 provide more insight into the load volatility of the house profiles using box-plots statistics [50]. The boxplots provide

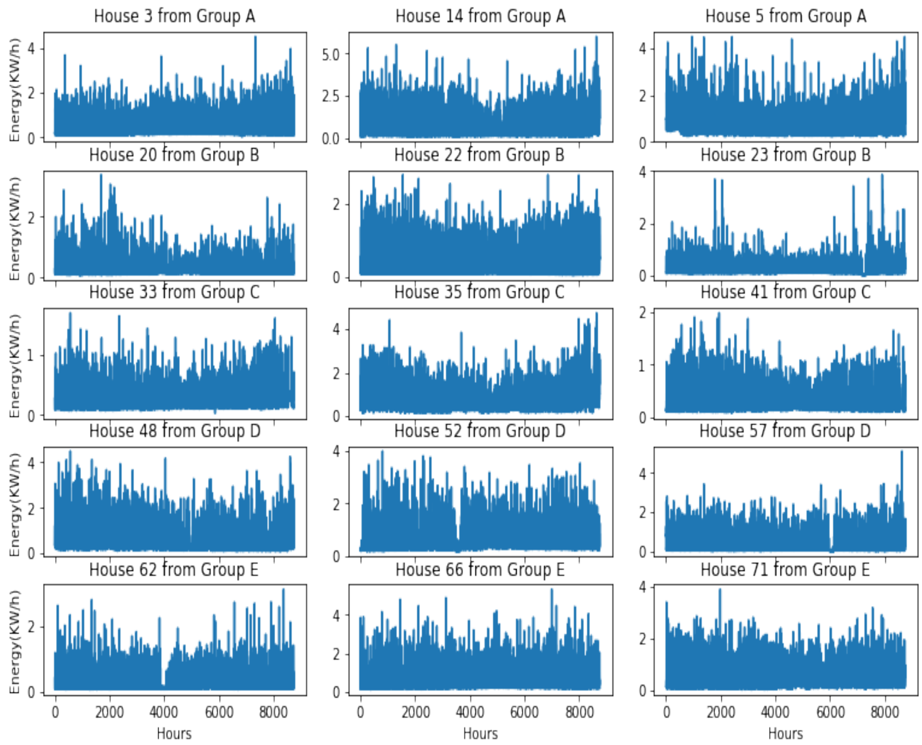

Figure 4: Hourly energy consumption of sample houses in different groups over one year (2013)

information on the median value and variability of the consumption values. Figure 5 shows how the hourly energy load of one house changes during different days of a week. For example, this house has experienced high variations in hourly consumption during the weekend and on Thursday than the other weekdays. Moreover, the median value of energy consumption has been increased over the weekend. The bobbles in the plot indicate that a few consumption values are out of the maximum range; $Q3 + 1.5 \times (Q3$ - $Q1)$ [50] which can be considered as outliers. For instance, on Saturday, there have been recordings higher than $1 + 1.5 \times (1.0$ - $0.3) = 2.05\,KW/h$ showing by bobbles.

Figure 6 similarly illustrates the hourly load volatility of houses in each customer group over one working day. We can see that there

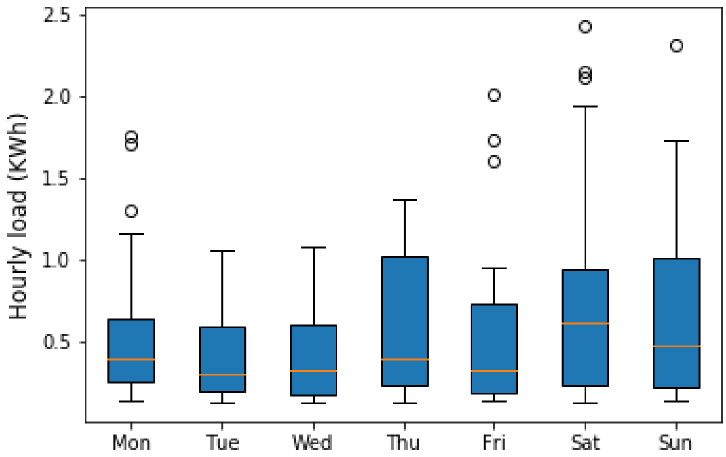

Figure 5: Boxplot statistics for House 33 over one week. (4. March to 11. March 2013)

are certain houses where their hourly consumption patterns change very little, such as houses 3, 4 and 8 in group A; potentially easier to predict while there are some houses (such as 15 in group A or 18 in group B) experiencing major changes in their hourly load profiles which can make them difficult to forecast.

Table 1 provides summary statistics of house profiles per customer group for the whole test period. The buildings in group A on average, have the highest mean electricity load over different time slots (hour, day and week) with the highest deviations from the mean values (standard deviation values of 0.72, 6.5 and 34). Similarly, customers in group D consume high energy but with lower deviations. On the other hand, the customers in groups B and C behave similarly and on average consume less electricity over a year. However, the lowest average values and smallest volatilities are recorded for the households in group E.

7 Forecasting Experiments and Results

In this work, we performed two separate experiments; one for developing and fine-tuning of the models on the validation set and

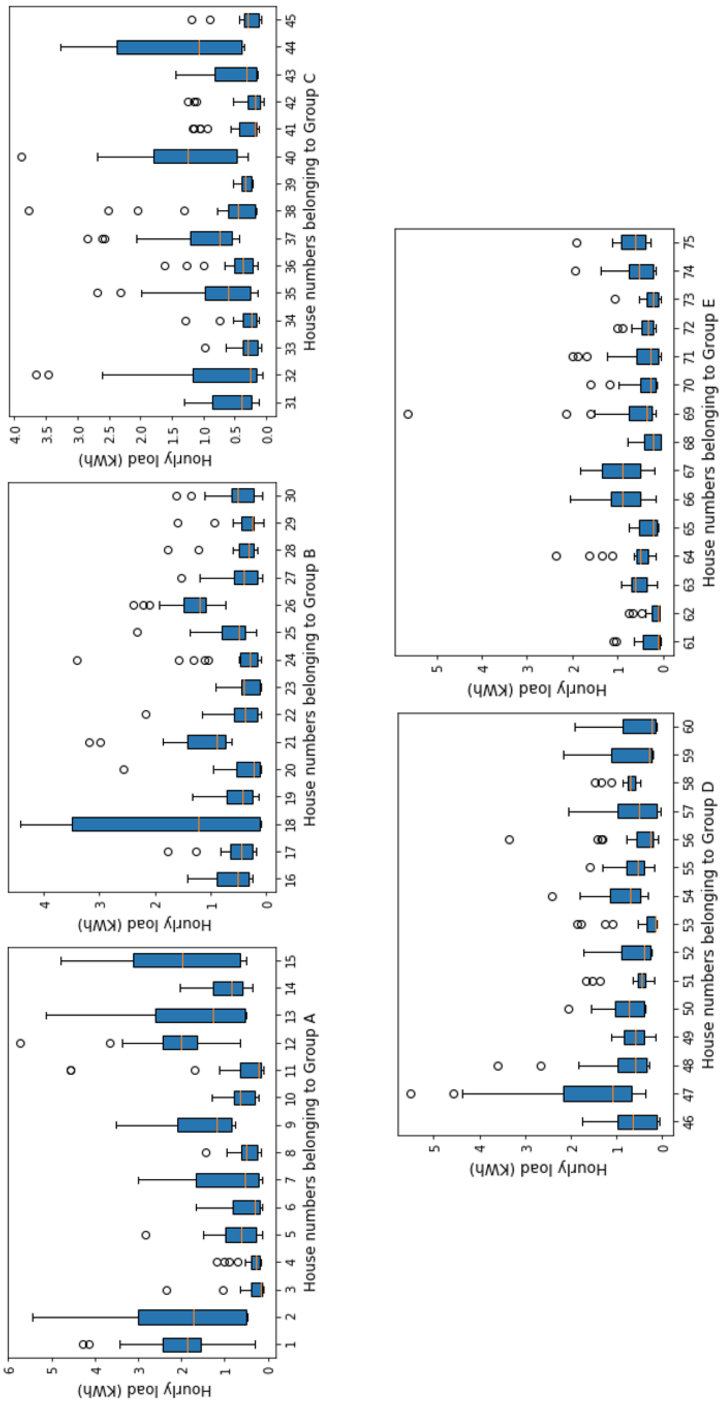

Figure 6: Boxplot statistics for 75 houses in a working day 2013 March 25

Table 1: Descriptive statistics of the dataset

User Group	Block Number	Number of houses	(Average Value Over a Group in KW/h) Mean and SD. of hourly consumption	Mean and SD. of daily consumption	Mean and SD. of weekly consumption	Total consumption over 1 year
Group A	B0	15	0.87 (± 0.72)	21.3 (± 6.51)	145 (± 34.3)	7707
Group B	B3	15	0.53 (± 0.45)	12.28 (± 3.99)	88.4 (± 21.3)	4689
Group C	B4	15	0.52 (± 0.42	12.7 (± 3.44)	88 (± 18.6)	4666
Group D	B10,B11	15	0.64 (± 0.52)	15.4 (± 4.57)	106 (± 24.2)	5627
Group E	B24,B27	15	0.44 (± 0.40)	10.7 (± 3.29)	74.2 (± 16.2)	3933

another for the performance evaluation on the test set. To do the experiments, all the time series models were implemented using Keras [2] and Scikit-learn [51] libraries. Total computations were conducted in Python 3 on a MacBook Pro machine @3,1 GHz, Intel Core i5 and 16 GB RAM.

The data set including 75 house profiles with hourly intervals was split into three separate subsets: trainset with 60% of data (45 houses), validation set with 20% of data (15 houses) and test set with the rest 20% (15 houses). The selection of validation and test sets was not performed randomly, instead, from each acorn group we selected 3 house profiles as the validation set and three house profiles as the test set with various levels of yearly consumption and average hourly load variations. This selection approach would help us to assess the model performance on a variety of house profiles with different statistical characteristics and consumption behaviors. The random selection might lead us to choose biased datasets either simple or complicated profiles which may cause overestimation or underestimation of the models' forecasting capabilities.

7.1 Model Development and Tuning

In this step, we implement the models based on the architecture design explained in Section 3 and tune the parameters according to Section 4. One of the aims of these experiments is to understand how the model's accuracy is affected by the size of training data and the number of input variables. The number of predictor variables and the size of input data can be influential on the performance of machine learning algorithms. If we build a model with an insufficient number of variables and training records, the model will be too simple and is not able to learn the relation between input and output variable(s). In contrast, the model complexity and computation time would increase if it is fed with too many features and redundant information.

As mentioned earlier, for all models, the input variables are considered as the load lags from previous time steps. The number of lags that were tested in our experiments varies from 1 to 9 e.g., the load

[2] https://keras.io

consumption for the previous 1 to 3 h and the load consumption at the previous 1 to 6 and 1 to 9 h. Regarding the size of the training set, four subsets of training data were considered for the evaluation: 25%, 50%, 75% and 100% of the total train size represented by D1, D2, D3, and D4 respectively. Figure 7 shows the evolution of the average prediction error on the validation set versus the number of input variables and the size of the training set for each of the four models studied.

According to the bar plots, the average prediction errors of all models decreased between 1% to 3% when the size of the training set increased from D1 to D3. However, a further increase in the training size has not affected the accuracy of all models similarly. The SVR and FFNN have shown the highest accuracy when they are trained on D3 dataset while GBRT and LSTM performed the best with the largest size of input data (D4).

Furthermore, the prediction error does not follow a clear trend regarding the number of input lags. For the GBRT and FFNN, it is observed that with the increase of the input number from 6 to 9 the errors tend to rise. This suggests that the hourly consumption pattern can be captured by the smaller number of input lags. Thus, in some cases, more lag can be discarded in order to reduce the computation cost and complexity of the model. The likely reason is that although the input to the model (past consumption variables) is highly correlated with the target variable (one-hour ahead consumption), they are also highly correlated with each other as they are consecutive lags. The mutual dependence between consecutive lags indicates redundancy of information they convey to the model which would not boost the learning ability, rather it could increase the training time.

However, for the LSTM on average more lag variables seem more informative to the model. For the SVR the forecasting error does not show a clear pattern concerning the input size.

7.2 Model Evaluation

In this section, we selected the final models (best variants) based on the minimum CWE obtained in the previous section. In the cases

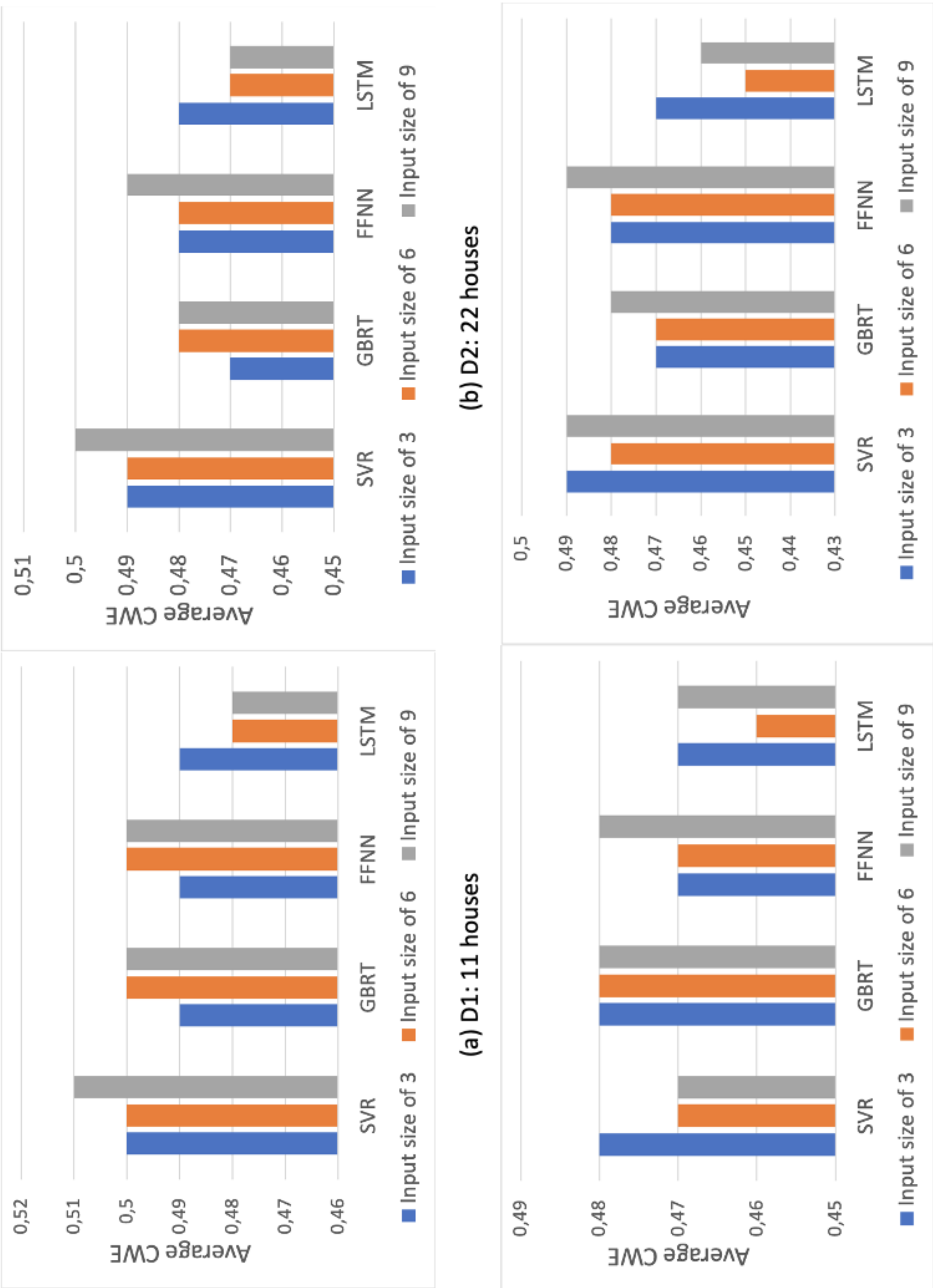

Figure 7: Error analysis with respect to the number of input variables and the size of the training set

Table 2: Characteristics of the best-trained models

Best variant	Training size	Training time (Minutes)	Parameters
SVR	33 houses	45	Kernel: RBF, C:10, Gamma: 0.001, Epsilon: 0.2, input: 6 load lags
GBRT	45 houses	15	Max Depth: 2, Learning rate: 0.06, n. estimatores: 300, n.features: 6, 6 load lags
FFNN	33 houses	30	Hidden layer: 1, Hidden neurons: 13, Optimizer: Adam, Weight.init mode: Golrot Normal, 6 load lags
LSTM	45 houses	40	LSTM layer: 1, LSTM cells: 10, Activation Function: ReLu Dropout rate: 0.1, 6 load lags

where the CWE results were the same, we picked the variant with smaller daily peak MAPE error. Table 2 provides information about the best variants; the size of training data, the number of input variables, the parameters and the training time.

The best variants are then tested on the test profiles. To understand the generalization ability of each model to different profiles, we computed the average error metrics over 15 houses. To estimate how much the error values, vary from the average, the corresponding standard deviation (SD) for each error metric and model is further reported. The lower standard deviation values for a model indicates a narrower range of errors and implicitly more robustness and consistency of the model. Table 3 reports the average forecasting errors for one-hour ahead predictions. The reported CWE values prove that, on average, the GBRT, FFNN and LSTM slightly outperform SVR in hourly load predictions.

Table 3: Average performance of best variants on 15 test houses

Average computed over predictions					
Model	RMSE ±SD (KW/h)	MAE ±SD (kW/h)	MASE ±SD (KW/h)	DpMAPE± SD (%)	CWE
SVR	0.36± 0.1	0.24± 0.05	1.12± 0.16	19.56± 3.68	0.48
GBRT	0.36± 0.1	0.23± 0.06	1.06± 0.12	17.88± 3.18	0.45
FFNN	0.35± 0.1	0.22± 0.06	1.01± 0.09	18.72± 4.08	0.44
LSTM	0.35± 0.1	0.21± 0.06	0.98± 0.07	17.76± 3.64	0.43

The AI-based models compared to the CART algorithm (GBRT) obtain better performance considering the average CWE (0.44 and 0.43 versus 0.45). However, GBRT and LSTM detect better the daily peak with an average DpMAPE of 17.8 and 17.7 KW/h respectively. In general, all models scale well and demonstrates robustness with average MAE of at most 0.24 KW/h and standard deviation of at most 0.06. Figures 8 and 9 illustrate how the models predicted the energy consumption of each test house over one week during the spring season.

Except for the SVR which slightly overestimates the real consumption values in most houses, the other three models mainly demonstrate a good match and steady weekly prediction for the one-hour ahead estimation. Figure 10 provides more insights into the variability or dispersion of error metrics which were averaged over all profiles and compared among different ML models.

The distribution of average DpMAPE errors proves that the median value of the peak prediction errors of GBRT and LSTM is less than the ones in other techniques. This means that these models adapt better to changes in daily peak consumption and achieve low perdition errors. The small size of the boxplot for the GBRT algorithm even shows the least variability in the peak errors, thus higher robustness.

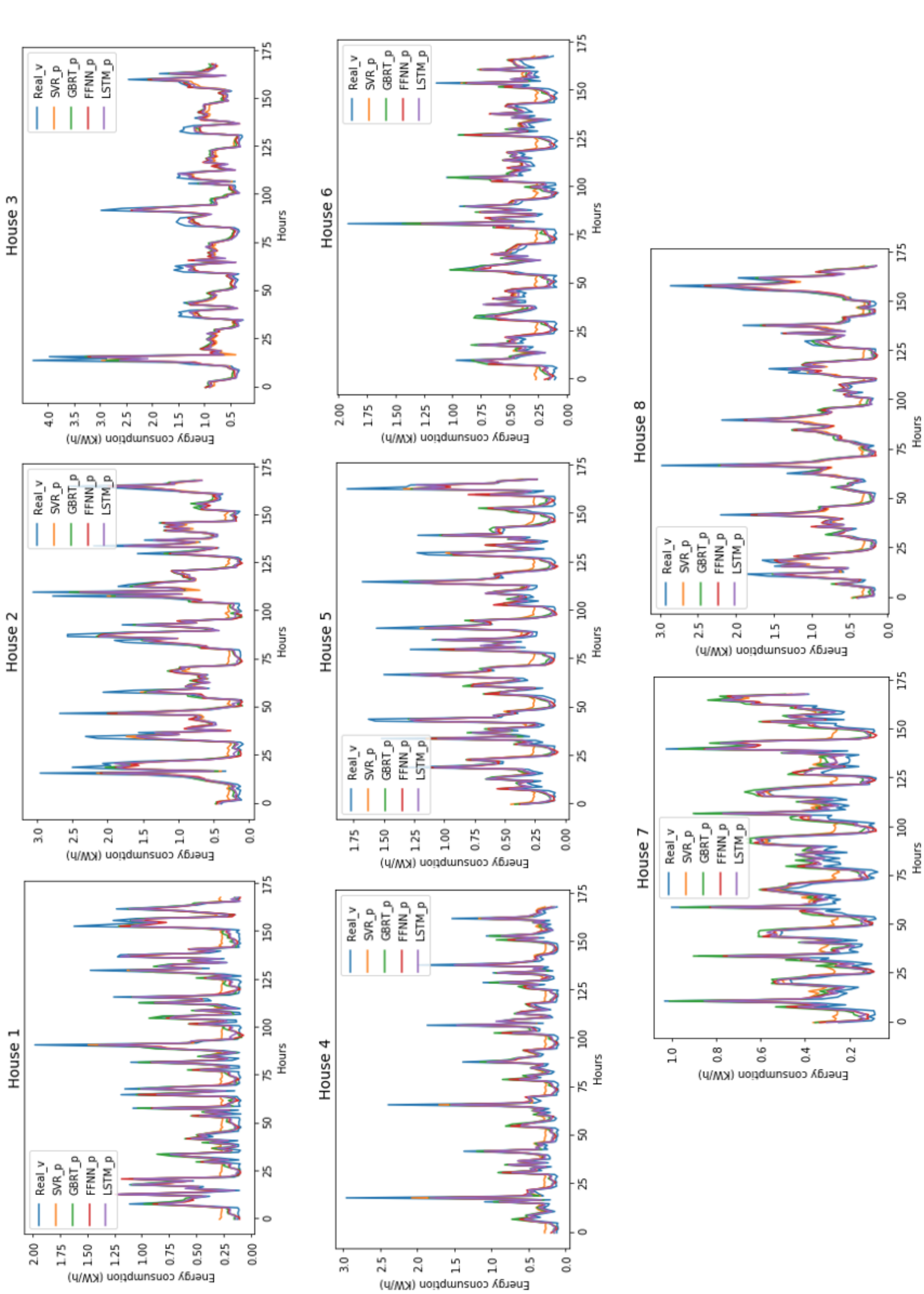

Figure 8: Real consumption of test houses (from number 1 to 8) versus predictions over one week

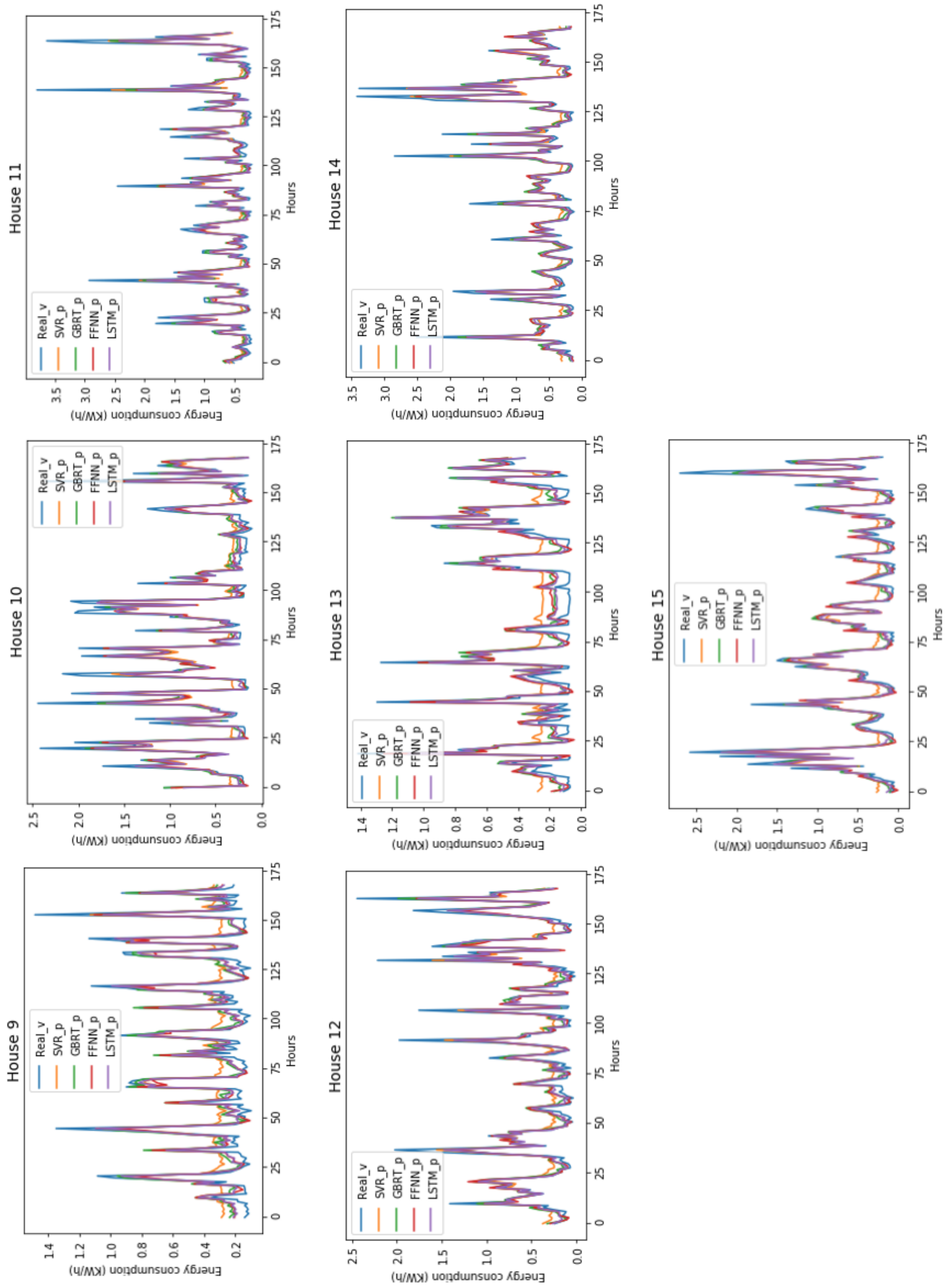

Figure 9: Real consumption of test houses (from number 9 to 15) versus predictions over one week

The ability of these models in obtaining high accuracy is also visible through boxplot statistics of average MASE. The median values in the plots of RMSE and MAE indicate that for all the models, half of the errors are around or less than 0.4 and 0.25 KW/h respectively. However, these plots are not informative enough for comparative analysis.

Another comparison is carried out by considering the diversity of consumer groups. Figure 11 shows the average mean absolute error in five distinct customer groups. The prediction error for the house profiles belonging to group A was the highest, while in the other categories it reduced between 2% and 15%. Groups B and C show the lowest values for different models out of all the other groups. It also confirms the lower average prediction error as FFNN, and LSTM were used as the prediction algorithms. The superiority of these models in hourly prediction is also visible for the test houses in other customer groups. Overall, we can conclude that the forecasting task in group A and group D is more challenging due to high average consumption over a year as well as high hourly load variations. The models in groups B and C, on the other hand, obtained higher accuracy due to the lower load volatility and lower average yearly consumption of the house profiles. These customers have shown the most predictable profiles. The interesting finding is that the consumption behavior of the houses belonging to group E with the lowest average consumption and the lowest daily variations were still difficult to predict. The probable reason is that the majority of houses in the training phase, on average, have experienced higher hourly energy consumption than these test profiles. Therefore, the trained models could not learn their consumption behavior properly.

The final analysis was performed to evaluate the effect of temperature and seasonal events on the prediction accuracy of various techniques. Fig.12 shows the average MAE of four techniques over all test houses in four consecutive seasons of the Year 2013. It is evident from the chart the SVR produced the highest error than the other three models over the seasons, though its estimation error reached the lowest (around 0.07 KW/h) in the summer. The other three models also predict summer load with the highest accuracy. The forecasting error during spring and autumn increased for the most

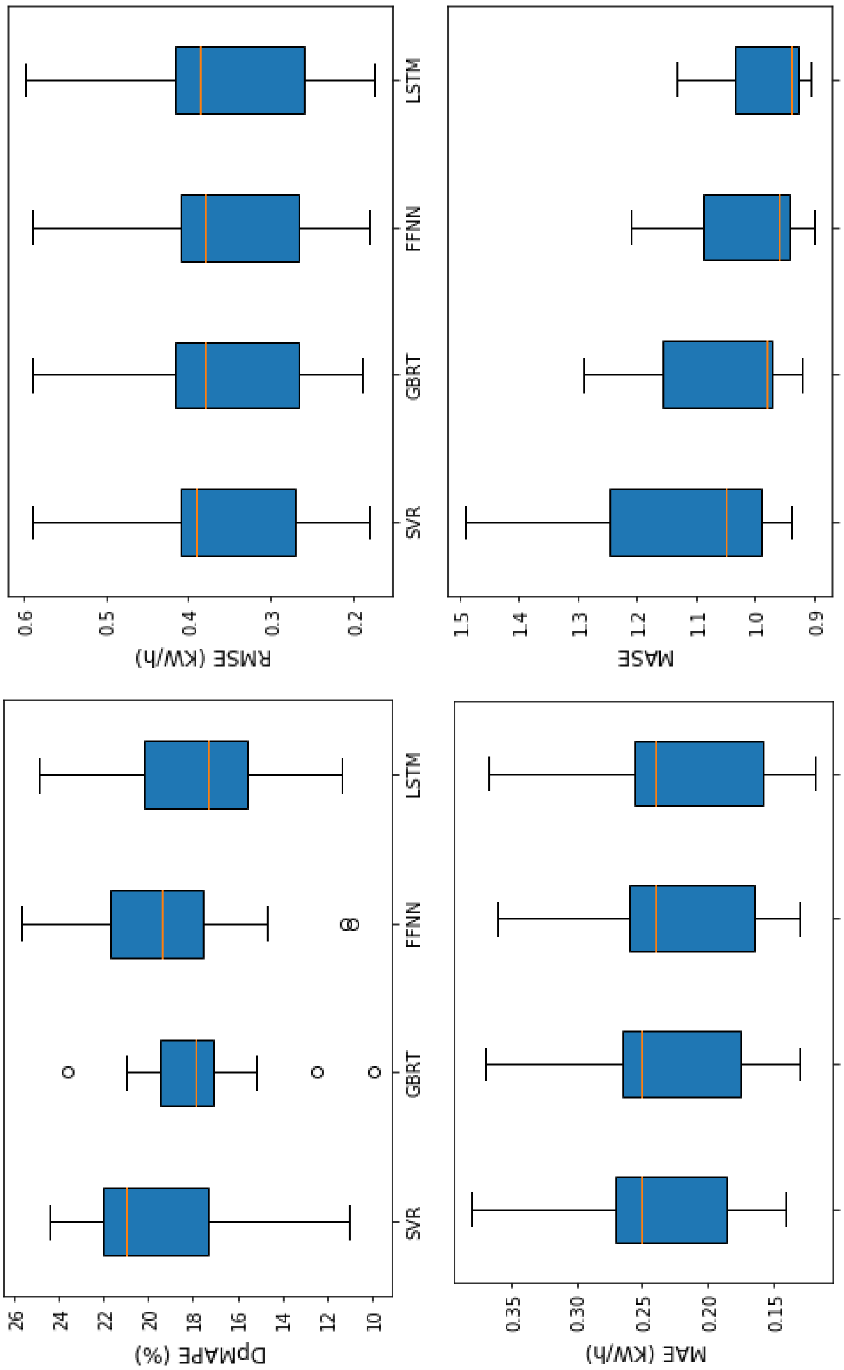

Figure 10: Comparison of models based on boxplot of forecasting error statistics

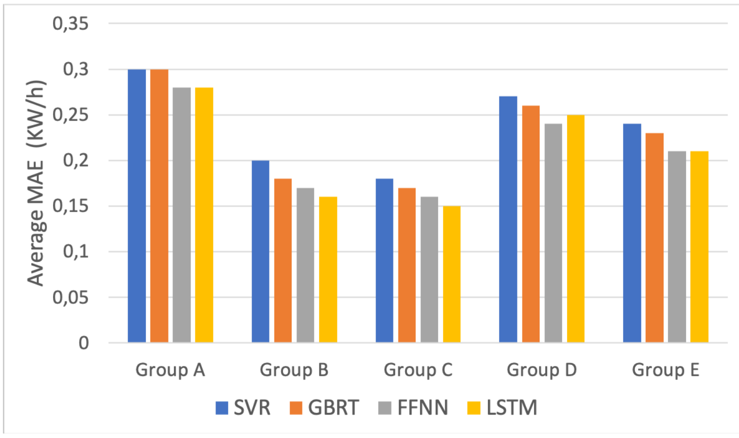

Figure 11: Comparison of models based on average Mean Absolute Error (MAE) per customer group

models by around 10%. In winter the SVR and GBRT had another 10% increase, while FFNN and LSTM seemed more adaptable to seasonal change and their error remained unchanged.

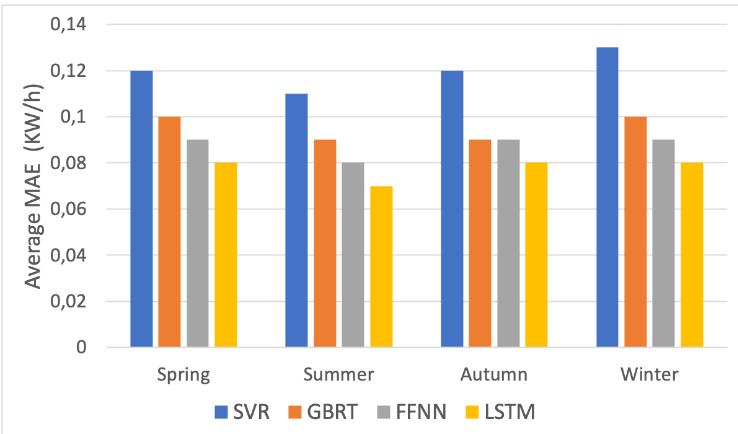

Figure 12: Comparison of models based on average MAE per season

8 Conclusions

This paper presents analysis and comparison of hour-ahead load forecasting with four data-driven models known as SVR, GBRT, FFNN and LSTM. They were trained on historical load data provided by the UK residential smart meters. Their generalization ability was evaluated on the house profiles which were not previously trained on. The test houses were chosen from five customer groups with different levels of load volatility and average yearly consumption. The sensitivity of each algorithm was also tested to the number of training houses and the number of input lag variables. The main findings are summarized as follows:

(1) Although the models were merely fed with consumption values as the predictors, in general, they could provide stable one-hour ahead prediction over a one-year period.

(2) The AI methodologies; LSTM and FFNN compared to the other two techniques, adapted better to changes while performing predictions, following the trend of real consumption and on average, achieving lower prediction errors.

(3) . With regard to daily peak load estimations of various profiles, the GBRT in addition to LSTM outperformed other techniques.

(4) As for the computation cost, the GBRT is the fastest algorithm to be trained and fine-tuned among the others. On the contrary, the training times of the SVR and LSTM are significantly high, especially when the training size for the SVR grows or the number of variables for tuning in the LSTM network increases.

(5) Increasing the number of training houses could improve the accuracy of forecasts as long as the additional profile(s) raise the model knowledge about the test profile. This implies a larger training set does not necessarily boost the model performance.

(6) Increasing the number of inputs does not have a similar effect on the performance of different variants. Some models perform better with recent information about past consumption and some need more knowledge on past consumption values.

(7) A comparative study among the five customer groups shows that the customers with the lower average amount of yearly consumption and smaller hourly load volatility generate more predictable profiles.

(8) An analysis of seasonal predictions reveals that the seasons with lower temperatures usually come with more load violations, thus making forecasting more difficult for almost all models.

Future lines of research in short-term load forecasting at the individual building level aim to customize forecasting techniques for the consumer groups with large variations in their consumption patterns. performance evaluation of the studied techniques on the dataset with higher resolution and for longer forecasting horizons can also be another research direction for the future work.

References

[1] Department for Business Energy and UK Industrial Strategy: London. *Smart Meters, Quarterly Report to end December 2016*. URL: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/512048/2015_Q4_Smart_Meters_Report.pdf (visited on 03/30/2017).

[2] U.S. Energy Information and Administration. *How Many Smart Meters are Installed in the United States, and Who Has Them?* URL: https://www.eia.gov/tools/faqs/faq.php?id=108&t=3 (visited on 01/28/2020).

[3] Y. Wang, Q. Chen, T. Hong, and C. Kang. "Review of Smart Meter Data Analytics: Applications, Methodologies, and Challenges." In: *IEEE Transactions on Smart Grid* 10.3 (2019), pp. 3125–3148. DOI: 10.1109/TSG.2018.2818167.

[4] Mina Farmanbar, Kiyan Parham, Øystein Arild, and Chunming Rong. "A Widespread Review of Smart Grids Towards Smart Cities." In: *Energies* 12.23 (2019). ISSN: 1996-1073. URL: https://www.mdpi.com/1996-1073/12/23/4484.

[5] Seunghyoung Ryu, Jaekoo Noh, and Hongseok Kim. "Deep neural network based demand side short term load forecasting." In: *Energies* 10.1 (2017), p. 3.

[6] Elena Mocanu, Phuong H Nguyen, Madeleine Gibescu, and Wil L Kling. "Comparison of machine learning methods for estimating energy consumption in buildings." In: *2014 international conference on probabilistic methods applied to power systems (PMAPS)*. IEEE. 2014, pp. 1–6.

[7] A. M. Pirbazari, A. Chakravorty, and C. Rong. "Evaluating Feature Selection Methods for Short-Term Load Forecasting." In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2019, pp. 1–8. DOI: 10.1109/BIGCOMP.2019.8679188.

[8] Hai-xiang Zhao and Frédéric Magoulès. "A review on the prediction of building energy consumption." In: *Renewable and Sustainable Energy Reviews* 16.6 (2012), pp. 3586–3592.

[9] Luis Pérez-Lombard, José Ortiz, and Christine Pout. "A review on buildings energy consumption information." In: *Energy and buildings* 40.3 (2008), pp. 394–398.

[10] Rob Hyndman, Anne B Koehler, J Keith Ord, and Ralph D Snyder. *Forecasting with exponential smoothing: the state space approach*. Springer Science & Business Media, 2008.

[11] Ahmad S Ahmad, Mohammad Y Hassan, Md P Abdullah, Hasimah A Rahman, F Hussin, Hayati Abdullah, and Rahman Saidur. "A review on applications of ANN and SVM for building electrical energy consumption forecasting." In: *Renewable and Sustainable Energy Reviews* 33 (2014), pp. 102–109.

[12] Mehdi Khashei and Mehdi Bijari. "An artificial neural network (p, d, q) model for timeseries forecasting." In: *Expert Systems with applications* 37.1 (2010), pp. 479–489.

[13] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks." In: *arXiv preprint arXiv:1409.3215* (2014).

[15] Franklin L Quilumba, Wei-Jen Lee, Heng Huang, David Y Wang, and Robert L Szabados. "Using smart meter data to improve the accuracy of intraday load forecasting considering customer behavior similarities." In: *IEEE Transactions on Smart Grid* 6.2 (2014), pp. 911–918.

[16] Magnus Dahl, Adam Brun, Oliver S Kirsebom, and Gorm B Andresen. "Improving short-term heat load forecasts with calendar and holiday data." In: *Energies* 11.7 (2018), p. 1678.

[17] W. Kong, Z. Y. Dong, D. J. Hill, F. Luo, and Y. Xu. "Short-Term Residential Load Forecasting Based on Resident Behaviour Learning." In: *IEEE Transactions on Power Systems* 33.1 (2018), pp. 1087–1088. DOI: 10.1109/TPWRS.2017.2688178.

[18] Nelson Fumo and MA Rafe Biswas. "Regression analysis for prediction of residential energy consumption." In: *Renewable and sustainable energy reviews* 47 (2015), pp. 332–343.

[19] Sp Pappas, L Ekonomou, D Ch Karamousantas, GE Chatzarakis, SK Katsikas, and Liatsis. "Electricity demand loads modeling using ARMA models." In: *Energy* 33 (2008), pp. 1353–1360.

[20] Xiaoshu Lü, Tao Lu, Charles J Kibert, and Martti Viljanen. "Modeling and forecasting energy consumption for heterogeneous buildings using a physical–statistical approach." In: *Applied Energy* 144 (2015), pp. 261–275.

[21] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control.* John Wiley & Sons, 2015.

[22] UVBR Thissen, R Van Brakel, AP De Weijer, WJ Melssen, and LMC Buydens. "Using support vector machines for time series prediction." In: *Chemometrics and intelligent laboratory systems* 69.1-2 (2003), pp. 35–49.

[23] Ali Lahouar and J Ben Hadj Slama. “Day-ahead load forecast using random forest and expert input selection.” In: *Energy Conversion and Management* 103 (2015), pp. 1040–1051.

[24] M Hambali, M Akinyemi, and N JYusuf. “Electric power load forecast using decision tree algorithms.” In: *Comput. Inf. Syst. Dev. Inform. Allied Res. J* 7.4 (2016), pp. 29–42.

[25] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. “An empirical comparison of machine learning models for time series forecasting.” In: *Econometric Reviews* 29.5-6 (2010), pp. 594–621.

[26] Hamid R Khosravani, Marıa Del Mar Castilla, Manuel Berenguel, Antonio E Ruano, and Pedro M Ferreira. “A comparison of energy consumption prediction models based on neural networks of a bioclimatic building.” In: *Energies* 9.1 (2016), p. 57.

[27] K. Yang and L. Zhao. “Application of Mamdani Fuzzy System Amendment on Load Forecasting Model.” In: *2009 Symposium on Photonics and Optoelectronics*. 2009, pp. 1–4. DOI: 10.1109/SOPO.2009.5230275.

[28] DK Chaturvedi, R Kumar, M Mohan, and PK Kalra. “Artificial neural network learning using improved genetic algorithm.” In: *J. IE (I), EL* 82 (2001).

[29] Arjun Baliyan, Kumar Gaurav, and Sudhansu Kumar Mishra. “A review of short term load forecasting using artificial neural network models.” In: *Procedia Computer Science* 48 (2015), pp. 121–125.

[30] Daniel L Marino, Kasun Amarasinghe, and Milos Manic. “Building energy load forecasting using deep neural networks.” In: *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2016, pp. 7046–7051.

[31] Weicong Kong, Zhao Yang Dong, Youwei Jia, David J Hill, Yan Xu, and Yuan Zhang. “Short-term residential load forecasting based on LSTM recurrent neural network.” In: *IEEE Transactions on Smart Grid* 10.1 (2017), pp. 841–851.

[32] Rahul Kumar Agrawal, Frankle Muchahary, and Madan Mohan Tripathi. "Long term load forecasting with hourly predictions based on long-short-term-memory networks." In: *2018 IEEE Texas Power and Energy Conference (TPEC)*. IEEE. 2018, pp. 1–6.

[33] Tae-Young Kim and Sung-Bae Cho. "Predicting residential energy consumption using CNN-LSTM neural networks." In: *Energy* 182 (2019), pp. 72–81.

[34] Abdullah Al Mamun, Muntasir Hoq, Eklas Hossain, and Ramazan Bayindir. "A hybrid deep learning model with evolutionary algorithm for short-term load forecasting." In: *2019 8th International Conference on Renewable Energy Research and Applications (ICRERA)*. IEEE. 2019, pp. 886–891.

[35] Salah Bouktif, Ali Fiaz, Ali Ouni, and Mohamed Adel Serhani. "Optimal deep learning lstm model for electric load forecasting using feature selection and genetic algorithm: Comparison with machine learning approaches." In: *Energies* 11.7 (2018), p. 1636.

[36] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[37] Ming-Wei Chang, Bo-Juen Chen, and Chih-Jen Lin. "EUNITE network competition: Electricity load forecasting." In: *National Taiwan University*. Citeseer. 2001.

[38] Jerome H Friedman. "Stochastic gradient boosting." In: *Computational Statistics & Data Analysis* 38.4 (2002), pp. 367–378.

[39] María del Carmen Ruiz-Abellón, Antonio Gabaldón, and Antonio Guillamón. "Load Forecasting for a Campus University Using Ensemble Methods Based on Regression Trees." In: *Energies* 11.8 (2018).

[40] Cong Feng, Mingjian Cui, Bri-Mathias Hodge, and Jie Zhang. "A data-driven multi-model methodology with deep feature selection for short-term wind forecasting." In: *Applied Energy* 190 (2017), pp. 1245–1257.

[41] Aurélien Géron. “Hands-on machine learning with scikit-learn and tensorflow: Concepts.” In: *Tools, and Techniques to build intelligent systems* (2017).

[42] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. “Learning to forget: Continual prediction with LSTM.” In: (1999).

[43] R. Lippmann. “An introduction to computing with neural nets.” In: *IEEE ASSP Magazine* 4.2 (1987), pp. 4–22. DOI: 10.1109/MASSP.1987.1165576.

[44] Geoffrey E Hinton. “Rectified linear units improve restricted boltzmann machines vinod nair.” In: (2010).

[45] *Types of Optimization Algorithms used in Neural Networks and Ways to Optimize Gradient Descent*. URL: https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6 (visited on 03/21/2020).

[46] Rob J Hyndman and Anne B Koehler. “Another look at measures of forecast accuracy.” In: *International journal of forecasting* 22.4 (2006), pp. 679–688.

[47] B. Asare-Bediako, W. L. Kling, and P. F. Ribeiro. “Day-ahead residential load forecasting with artificial neural networks using smart meter data.” In: *2013 IEEE Grenoble Conference*. 2013, pp. 1–6. DOI: 10.1109/PTC.2013.6652093.

[48] *SmartMeter Energy Consumption Data in London Households—London Datastore*. URL: https://data.london.gov.uk/dataset/smartmeter-energy-use-data-in-london-households (visited on 02/10/2020).

[49] *ACORN*. URL: https://acorn.caci.co.uk/downloads/Acorn-User-guide.pdf (visited on 03/23/2013).

[50] Vladimir N Vapnik. “An overview of statistical learning theory.” In: *IEEE transactions on neural networks* 10.5 (1999), pp. 988–999.

[51] Fabian Pedregosa and et al. Varoquaux. “Scikit-learn: Machine Learning in Python.” In: *J. Mach. Learn. Res.* 12 (2011), pp. 2825–2830.

Paper III:
Improving Load Forecast Accuracy of Households Using Load Disaggregation Techniques 2020.

A. M. Pirbazari, M. Farmanbar, A. Chakravorty and C. Rong, "Improving Load Forecast Accuracy of Households Using Load Disaggregation Techniques," *2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom)* and *IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), 2020*, pp. 843-851, doi: 10.1109/iThings-GreenCom-CPSCom-SmartData-Cybermatics50389.2020.00140.

This paper is not included in Brage due to copyright restrictions.

Paper IV:
An Ensemble Approach for Multi-step Ahead Energy Forecasting of Household Communities

An Ensemble Approach for Multi-step Ahead Energy Forecasting of Household Communities

4036 Stavanger, Norway
A.M. Pirbazari[1], E. Mohan[2], A. Chakravorty[1], W. Elmenreich[2], C. Rong[1]

[1] Department of Electrical Engineering and Computer Science, University of Stavanger
[2] Institute of Networked and Embedded Systems, University of Klagenfurt, Klagenfurt, Austria

Abstract:
This paper addresses the estimation of household communities' overall energy usage and solar energy production, considering different prediction horizons. Forecasting the electricity demand and energy generation of communities can help enrich the information available to energy grid operators to better plan their short-term supply. Moreover, households will increasingly need to know more about their usage and generation patterns to make wiser decisions on their appliance usage and energy-trading programs. The main issues to address here are the volatility of load consumption induced by the consumption behaviour and variability in solar output influenced by solar cells specifications, several meteorological variables, and contextual factors such as time and calendar information. To address these issues, we propose a predicting approach that first considers the highly influential factors and, second, benefits from an ensemble learning method where one Gradient Boosted Regression Tree algorithm is combined with several Sequence-to-Sequence LSTM networks. We conducted experiments on a public dataset provided by the Ausgrid Australian electricity distributor collected over three years. The proposed model's prediction performance was compared to those by contributing learners and by conventional ensembles. The obtained results have demonstrated the potential of the proposed predictor to improve short-term multi-step forecasting by providing more stable forecasts and more accurate estimations under different day types and meteorological conditions.

1 Introduction

The shift towards low carbon and sustainable energy production is gaining momentum to support the increasing energy demand. Due to this transition, the installed PV generation capacity is expected to increase by more than 21.9 TW by 2050 [1]. Photovoltaics nowadays enables the generation of localized electricity among residential consumers at a lower cost than that of the power grid. Costs are even smaller if energy storage devices are used. Therefore, self-consumption; the consumption from self-produced electricity is expected to grow among households. Moreover, energy use nowadays in residential buildings is on the increase by using more electric vehicles and high demand appliances. In such an environment, forecasting energy demand and supply from micro-generation sources become necessary to tackle the instability induced by the integration of PV to the power grid and reduce the uncertainty of demand [2].

For electricity suppliers, forecasting demand and micro-generation provide useful information to achieve demand and supply equilibrium, serve peak demands, and maintain reliable grid operation. From the customers' point of view, energy forecasts through a smart Energy Management System (EMS), will enable them to make smarter decisions on managing their use, increasing self-consumption, trading energy and reducing electricity bills. Intelligent Energy management in buildings will lead to a decrease in the electricity intake from the power grid, which in turn lowers the total operating costs [3].

Regarding the forecast horizon, energy forecasts are made with various time scales corresponding to a particular decision-making activity. Very short-term (from a few minutes to a few hours ahead) is generally used for flow control and real-time dispatch; short-term (from a few hours to a few weeks ahead) for adjusting generation and demand and electricity trading; medium-term and long-term (from a few months to a few years ahead) for PV plant planning, power maintenance, etc.[4].

Generally, forecasting energy consumption with short-term to medium-term horizons at smaller scales such as a residential building or community level is quite challenging due to several demographic and economic factors which influence the load with different degrees.

These factors normally include population, size and structure of the building, number of residents, number of appliances under usage, heating, ventilation, air conditioning system, and weather data (humidity, wind speed, temperature, precipitation, etc.). A comprehensive study of primary features that influence electricity energy demand is conducted in [5].

Similarly, the PV power output is difficult to predict with short-term horizons due to its dependency on uncertain meteorological factors, such as solar irradiance, atmospheric temperature, module temperature, wind pressure, wind direction, and humidity. This causes the power output of a PV system to change dynamically. A recent correlation study reported in [6] shows that solar irradiance has the highest correlation with PV power output compared to other weather parameters. The result is validated for various weather conditions in [7] and [8].

The energy estimation can be grouped into two categories according to forecasting steps: one step-forecasting, which estimates future demand or supply one step ahead in time, and multi-step forecasting that predicts multiple time steps into the future. Several architectures are proposed in the literature for one to multi-step ahead energy forecasting at building levels. They are broadly categorised into three categories: physical, data-driven (statistical or computational intelligent) and hybrid methods [9] and [10]. Physical approaches also known as analytical methods rely on the mathematical modelling of the building under study. Data-driven methods, in contrast, focus on statistical analysis performed on historical time-series data with different input variables. Hybrid approaches incorporate both physical and data-driven methods to exploit the benefits of each approach.

There are many studies which have analysed the potential of data-driven and hybrid methods for electric load forecasting of buildings and cities at multistep ahead. The approaches based on Artificial Neural Network (ANN) [11], [12] and Support Vector Machines (SVMs) are successfully applied for energy analysis of buildings [13]. Moreover, the deep learning approaches have been utilized for energy consumption prediction at multistep ahead. A deep learning method based on 2D Convolutional Neural Network (CNN) to forecast one-day ahead load with the fifteen-minute resolution is investigated in [14]. The prediction accuracy of Auto-Regressive Integrated Moving Average (ARIMA), Long-Short-Term-Memory (LSTM), and Recurrent Neural Networks (RNN) models are compared in [15]. The results showed the effectiveness of the LSTM in comparison with ARIMA for multi-step electric load forecasting. A variant of LSTM based on Multi-Channel with time location (TL-MCLSTM) is proposed for multi-step short-term consumption forecasting in [16]. The results showed that their proposed method outperformed the compared methods including LSTM and CNN-LSTM.

Yang et.al in [17] investigated the potential of a hybrid model for multi-step load forecasting based on Extreme Learning Machine (ELM), Recurrent Neural Network (RNN), SVM and Multi-Objective Particle Swarm Optimization algorithm (MOPSO). The experiments on different cities showed that the optimization technique can improve the performance of the hybrid method and the combination technique can improve the prediction accuracy. Another ensemble approach based on Generalized Recurrent Neural Network (GRNN) and SVM is proposed in [18] to predict the one week ahead electricity demand of state loads. The experimental findings indicate that the proposed approach is highly effective in terms of prediction accuracy and model robustness.

The studies related to multi-step ahead PV production forecasting also reveal the strength of using Deep Learning (DL) and hybrid models. Various structures of LSTM networks in [19] and [20] have been proposed for solar power forecasting. Lee et al. [21] and Alzahrani et.al [22] have demonstrated the superior performance of deep models over conventional techniques for solar irradiance estimation. The authors in [10] have shown that the ensemble approaches in PV output forecasting increase the precision and efficiency of models compared with individual models by integrating linear and non-linear techniques.

A hybrid learning algorithm incorporating Self-Organizing maps (SO), Support Vector Regression (SVR), and Particle Swarm Optimization (PSO) is also presented in [23] to forecast hourly solar irradiance at city levels. It is found that the combined technique outperforms conventional forecasting models. Another comparative study in [24] shows the superiority of a blended model against individual algorithms including SVR, Random Forests (RF), Deep Neural Networks (DNN) and Extreme Gradient Boosting (XGB) for day-ahead PV output forecasting. In [25] the tree-based ensemble methods based on extra trees (ET) and random forests (RF) also demonstrate satisfactory results compared with support vector regression (SVR) as the widely used machine learning method.

Despite several studies proposed in the literature dedicated to short-term and multi-step forecasting, a few have investigated the potential of deep-learning-based hybrid techniques using various input variables. Presumably, no study has focused on forecasting both electricity consumption and PV production across one dataset at local level. Furthermore, most of the existing studies related to PV production forecasting have been conducted on a proprietary dataset. The limited availability of these datasets does not allow a fair comparison between the results obtained using different forecasting model architectures. Hence configuring the forecasting models on a specific dataset is not optimal for a similar problem or dataset. This also makes it difficult to reproduce the result. To close the research gap this paper presents four main contributions.

(1) An ensemble approach with two levels is proposed to develop forecasting models for energy consumption and energy generation of household communities at multi-steps ahead. In the first level, multiple forecasting algorithms as base learners predict both target outputs in one-step forward. In the second level, the predictions for each target are used to train a meta learner aimed at generating multi-step predictions separately for each target.

(2) To create a diverse ensemble, we choose one promising algorithm from the category of deep neural networks and one from the class of conventional ML algorithms. Instead of fine-tuning one deep network , multiple deep networks with different parameter settings were trained on the dataset and their forecasts were combined to create a more robust estimate.

(3) Three influential factors are considered as input variables to the forecasting models: time variables, meteorological data, and

historical electricity consumption and PV power output. Before model development, two feature selection techniques along with two machine learning algorithms are used to select the optimal subset of input variables.

(4) A publicly available dataset with actual observations from an Australian electricity distributor is selected as a case study with both electricity and solar PV output measurements from 300 houses. The forecasting methods are evaluated from the perspective of accuracy and prediction stability.

The paper is organized as follows: Section 2 presents and discusses detailed forecasting framework steps. Section 3 describes the case study, followed by forecasting experiments and results in section 4. Section 5 concludes the study.

2 Forecasting Framework

Fig. 1 depicts the framework for multi-step ahead energy forecasting, consisting of three main steps. Step One; data preprocessing, Step Two; model development and selection of the most accurate models, and Step Three; development and evaluation of an ensemble model based on the results of the previous step.

The first step is further classified into five main tasks: visual exploration, data cleaning, feature extraction and transformation, feature selection and input/output modelling. During the first step, we aim to better understand the data, improve the data quality, determine predictive features, select the most useful ones and convert data into the appropriate format for forecasting models. In the second step, several commonly used algorithms in time series forecasting e.g. ARIMA, SVR, LSTM etc. are trained and evaluated on large sets of training and validation data. The aim is to shortlist the most promising models for energy forecasting problem.

In the third step, the resulted algorithms from Step Two with the lowest prediction error on average i.e. Seq2Seq LSTM and GBRT, are combined to create an ensemble model. The trained ensemble technique is then applied to predict several household communities' energy

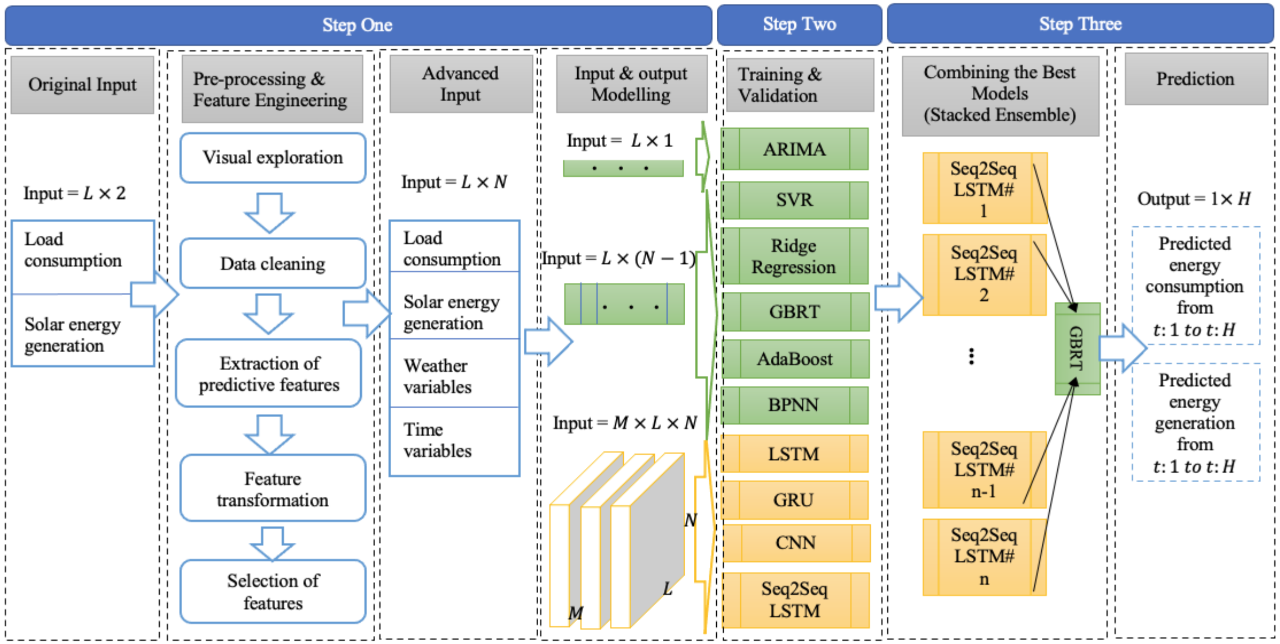

Figure 1: Framework for prediction of two targets: total energy consumption and energy generation. Original input data has the dimension of $L \times 2$ where L is the time lag and 2 refers to the number of target variables. The Advance input consists of N input variables, including energy consumption, energy generation, meteorological and time factors with the same number of lags L. For the models in green rectangles, except for ARIMA fitted to uni-variate data, the multivariate input was flattened to $L \times (N-1)$ array to meet the requirements of the input layer. In contrast, for the models represented with yellow rectangles, the input is transformed to $(M \times L \times N)$ tensor where M represents the batch size. The output dimension is 24-horizon ahead ($H = 24$). For the green models that do not support the prediction of two target variables in one-step forward, one regressor is fitted per target and for most of them that naturally do not support multi-step prediction, one regressor is fitted per time step. For the others, one algorithm is trained to predict multi-steps ahead of two output variables.

consumption and generation as test sets. The following subsections present the details of each step.

2.1 Step One

2.1.1 Visual Exploration

Visual exploration allows us to understand the dataset more effectively. It is also useful to recognize patterns and trends within the data more quickly. In this study, we provide several plots and statistics to serve these purposes. The results of visual exploration further promote our decisions on subsequent steps such as data cleaning, feature extraction, and feature engineering.

2.1.2 Data Cleaning

Typically, machine learning algorithms cannot perform effectively with missing features. The raw smart meter data may contain missing values due to transmission error or smart meter failures, which would degrade the data quality. Data cleaning helps in managing missed values in the time series of electricity load and solar output. To fix missed values, imputation, moving average (MA), and inference-based approaches can be used.

This study fills the missing values from the surrounding measurements with a variant of MA known as an exponential weighted moving average (EWMA) [26]. This technique solves the problem by computing the arithmetical mean of data surrounded at the two sides of missing value on condition of placing a higher weight on the latest data. As both electricity load and solar output continuously vary, two closer measurements are more similar. Thus, applying the EMWA technique can be useful for replacing missing values.

2.1.3 Feature Extraction

The performance of a forecasting model mainly depends on the input variables as the predictive features. As mentioned in Section 1, previous studies have shown several influential factors on accurate forecasting of household energy consumption and solar cells' output.

The most important and common factors between two targets include historical load measurements, weather conditions, time variables, and customer socio-economic factors.

In this analysis, we created a set of candidate features based on two references: the literature study and the results of data exploration. The majority of the candidate features are influential on both prediction targets such as outdoor temperature and historical load, while some are more effective for predicting only one target, such as 'solar zenith angle' for solar output estimation and 'weekends' or 'holidays' on electricity consumption prediction. Notably, the candidate set of features will be discussed in detail in Section 3, Subsection 2, and the subset of features as the final input to the forecasting models will be introduced in Section 3, Subsection 3.

2.1.4 Feature Transformation

Most machine learning algorithms perform well with numerical features instead of categorical features. In this work, the categorical attributes are transformed into numerical attributes using a commonly used method as One-hot encoding. This technique generates a binary feature for each subclass of categorical features. For instance, it converts a feature with two sub-classes into two binary features. Furthermore, some numerical features with the current scale or value are not informative enough as input to the forecasting models. Therefore, they are transformed into an appropriate format before using in the model development process. The attributes and functions used in the feature transformation process will be discussed in Section 3, Subsection 3.

2.1.5 Feature Selection

Selecting the best combination of the variables having a high correlation with energy consumption and production can improve the performance of the forecasting algorithm. However, an input space with redundancy and many inter-correlated features typically decreases the accuracy of the prediction model and contributes more to the over-fitting problem. To avoid overfitting, two feature selection methods, which have been proposed in the literature, including Pearson Correlation Coefficient (PCC) [27], [28] and Recursive Feature Elimination Technique (RFE) [29], [30], [31] were used to reduce the dimension of input space.

The PCC measures the linear correlation between two variables x and y as Equation (1):

$$\rho(x, y) = \frac{Cov(x, y)}{\sigma(x), \sigma(y)} \tag{1}$$

Where Cov is the covariance; $\sigma(x)$ and $\sigma(y)$ are the standard deviations of x and y. If we consider each candidate variable as x and each prediction target as y, then the x variables whose correlation with the y target exceed a predefined threshold can be selected as most related features. However, The PCC method cannot capture nonlinear relationships between the x and y variables. Therefore, the RFE method combined with a training algorithm was used to discover the variables with high prediction ability and even with nonlinear relations with the targets.

The RFE [4] ranks features to evaluate their importance according to a specific criterion. It also uses the model accuracy to determines the features contributing most to the prediction task in a recursive way. The determination process starts with training a prediction algorithm on the original feature set. It then continues with measuring the feature importance and removal of the less relevant ones. This procedure is repeated until it reaches the desired number of features to select . The results of the feature selection process are discussed in Section 3, Subsection 3.

2.1.6 Input and Output Modelling

As mentioned in Section 1, the task of energy load forecasting involves many influential factors. Each factor can be dependent on both its precedent values and the values of other factors. For instance, household energy usage and solar output are strongly correlated to their historical values and to air temperature values at the same

[4] Available at https://scikit-learn.org/stable/modules/feature_selection.html#rfe

time. To learn the potential correlations, it is useful for the learning algorithm to receive this information as a multivariate time series which contains multiple variable values at each observation time step. Therefore, in this analysis, we provide a description of multivariate energy forecasting. The aim is to use historical time series data to predict the future temporal values of multiple energy variables. The model input not only contains the historical energy factors, but also includes other predictive attributes. This problem is formulated as Equation (2):

$$D \xrightarrow{M(D)} (Y_1, Y_2) \tag{2}$$

Where $M(D)$ refers to the learning model which aims to predict the next H values of energy consumption and generation of households from time step t as two time series data $Y_1 = [y_{1,t}, y_{1,t+1}, \ldots, y_{1,t+H}]$ and $Y_2 = [y_{2,t}, y_{2,t+1}, \ldots, y_{2,t+H}]$ given a history multivariate time series dataset $D = (x_{i,j} | i = 1, 2, \ldots, N;\ j = t - L, \ldots, t - 1, t)$ where N denotes the number of input variables (features) and L represents the window length of history data.

2.2 Step Two

In this step, many forecasting models from different categories (e.g., Autoregressive, linear, SVM, Bagging and Boosting ensembles based on Decision trees, Neural networks, and deep learning) are trained and evaluated on a subset of training data using standard and suggested parameters in the literature. The models involved are briefly presented in the remainder of this section. The experimental settings and prediction results of given techniques will be discussed in Section 4.

2.2.1 Persistence

One of the easiest ways of forecasting a time series's future behavior is the so-called persistence model. Persistence in the context of energy forecasting assumes that the future values of the demand or supply are determined under the basis that the conditions stay constant between the current time and the future time. For long prediction periods, however, this strategy lacks the skill of forecasting. Persistence was only assessed here for comparative analysis.

2.2.2 Autoregressive Integrated Moving Average (ARIMA)

ARIMA belongs to the category of statistical models for forecasting time series data. In ARIMA, the generated forecasts are treated as a linear function of the most recent observations and past random errors. Mathematical details are provided in [32]. This technique is usually unable to capture non-linear relationships between components of the time series and is often applied to univariate time series data. Sample applications of the hybrid models with ARIMA applied to short-term time series forecasting can be found in [33] and [34].

2.2.3 Ridge Regression

Similar to the Linear Regression algorithm, Ridge regression assumes a linear relationship between input variables and the output. However, it makes the model simpler by adding L2 penalty to the loss function during training. L2 penalty has the effect of reducing coefficient values of those inputs that have less contribution to the forecasting task. It is calculated based on the sum of the squared coefficient values. In our research, Ridge regression [35] was selected to evaluate and compare the potential predictive ability of a regularized linear model against non-linear statistical models.

2.2.4 Support Vector Regression (SVR)

SVR is an extension of a Support Vector Machine (SVM) used for regression problems. The SVR is based on statistical learning theory and structural risk minimization. In this method, instead of minimizing the training error, the generalization error is reduced. The generalization functionality optimization is achieved by mapping the initial input space through non-linear kernel functions to a high-dimensional feature space. A mathematical explanation of SVR is provided in [36]. The SVR models are successfully applied to electrical load forecasting [37], [18], [38] as well as renewable energy prediction[39], [40], [41].

2.2.5 AdaBoost

AdaBoost is an ensemble method that combines many weak learners into a strong learner. The most common algorithm used with AdaBoost is one-level Decision Tree (DT) algorithm. In this method, the Decision Trees are sequentially added and trained as weak learners. This process repeats until a predefined number of learners have been created or there is no further reduction in the training error. In an AdaBoost used for regression tasks, final predictions are made by calculating the weighted median prediction of the learners in the ensemble. The detailed process is given in [42]. The benefit of AdaBoost over SVM is the ability to identify only those features with more predictive capacity during training. This ability would potentially lead to enhanced execution time due to lower dimensionality of input space.

2.2.6 Gradient Boosted Regression Tree (GBRT)

GBRT is another type of ensemble algorithm developed based on Boosting and Decision Tree algorithms. In this approach, the model loss is determined using a gradient descent technique when each weak learner is introduced to the GBRT ensemble. This process adds a tree to the model that decreases the loss. To improve the performance of the ensemble, each new learner's output is then added to the output of the sequences of the generated tree. The details of the algorithm is discussed in [43].

2.2.7 Back Propagation Neural Network (BPNN)

BPNN is a type of artificial neural network (ANN) which use a back-propagation algorithm [44] for training the network. The architecture of a BPNN model includes an input layer, one or more hidden layers and an output layer. The non-linear activation function in the hidden layer(s) can capture the complex relationship between variables in the input and output layers. The flow of signals in a conventional BPNN is from inputs to outputs. Thus the network architecture is called the Feed Forward network. The ability to estimate any continuous function, the high generalization ability, and imposing no restrictions on the input variables have made the ANNs considerably useful techniques for forecasting time series, specifically where the data volatility is high, as for example in load data. A review of using ANNs for building electrical energy consumption forecasting and photovoltaic power generation is provided in [45] and [46].

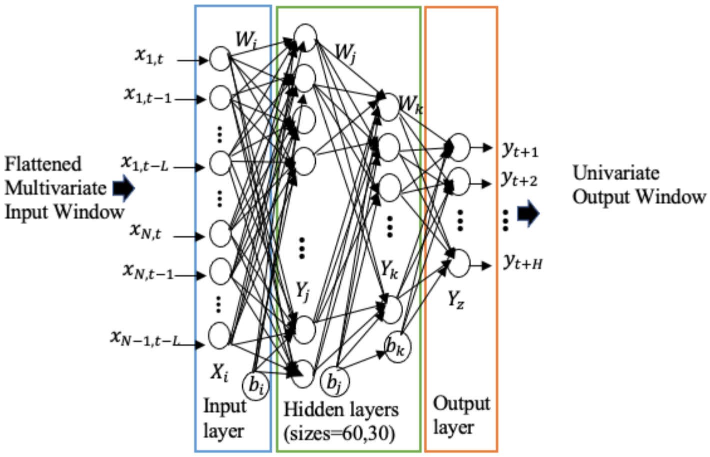

Figure 2: BPNN architecture for multi-step prediction

Fig. 2 depicts the architecture of BPNN, which was developed for our problem. To satisfy a BPNN network's input requirements, the time series data with $(N-1)$ variables from previous t time steps is first framed as sliding windows with the size of L. Reducing one variable from the input with the size of N indicates that the BPNN that is trained to predict only future load consumption values removes the solar energy generation variable from its input. Accordingly, the network which estimates solar output ignores electricity load from the input set. Each two-dimensional input window is then transformed to a flattened vector of X_i and is fed to the network where $X_i = (x_{1,t}, x_{1,t-1}, \ldots, x_{1,t-L}, ..., x_{N-1,t}, \ldots, x_{N-1,t-L})$ and $i = (1, 2, \ldots, (N-1) \times L)$. Next, the network computes an output vector of Y for the next time steps from $t : t+1$ to $t : t+H$, where H represent the forecast horizon. Each element of Y vector at time step t is then calculated by Equation (3):

$$y_t = b_k + \sum_{k=1}^{n_2} w_k\, f_2(\sum_{j=1}^{n_1} w_{jk}\, f_1(\sum_{i=1}^{p} w_{ij} + b_i) + b_j) \tag{3}$$

Where y_t is the output y of the BPNN at time t; f_1 and f_2 are the non-linear functions of the neurons in the first and second hidden layers; n_1 and n_2 are the number of neurons in the hidden layers; w_{ij} are the weights of neuron j connecting the input with the first hidden layer; and w_{jk} are the weights of neuron k connecting the neuron j in the first hidden layer with the neuron k in the second hidden layer; w_k are the weights connecting the output of the neuron k with the output neurons and b_i, b_j, b_k denote bias vectors in the input layer and two hidden layers respectively. Note that, in the given problem, one BPNN was trained per target for multi-step prediction. Both networks accept the same multivariate input data; however, one is trained to estimate energy consumption and one is trained to predict energy production.

2.2.8 Long-Short Term Memory network (LSTM)

LSTM, proposed initially by Hochreiter et.al [47], is an artificial recurrent neural network (RNN) [48] architecture that is well suited to time series prediction. There are feedback connections in the LSTM to update the state of neurons with previous inputs, in contrast to conventional feed-forward neural networks.

Moreover, unlike typical RNNs, they benefit from long-term memory cells to resolve the disadvantage of unstable gradients while learning series with long-term dependencies. There are four main connected layers in the basic LSTM cell, as shown in Fig. 3.

The main layer known as control state computes the long-term state c_t by analyzing the current input vector x_t and previous short-term state h_{t-1}. The other three layers are gate controller layers known as f_t for the forget gate, i_t for the input gate and o_t for the output gate. Gate operations, such as removal, writing and reading are performed to change the LSTM cell's states and its output at each time step. The LSTM computations are shown through Equation (4)

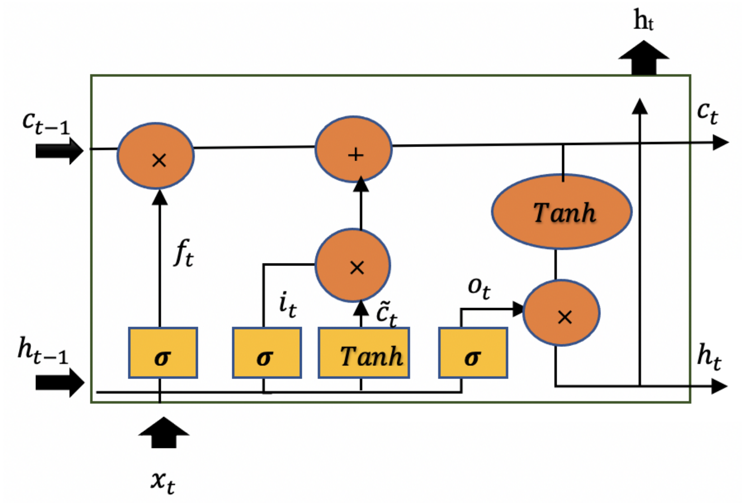

Figure 3: Structure of a basic LSTM cell

to Equation (9):

$$i_t = \sigma(W_{xi}^T.x_t + W_{hi}^T.h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_{xf}^T.x_t + W_{hf}^T.h_{t-1} + b_f) \tag{5}$$

$$o_t = \sigma(W_{xo}^T.x_t + W_{ho}^T.h_{t-1} + b_o) \tag{6}$$

$$\tilde{c}_t = \tanh{(W_{xc}^T.x_t + W_{hc}^T.h_{t-1} + b_c)} \tag{7}$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \tag{8}$$

$$h_t = o_t \otimes \tanh c_t \tag{9}$$

Where σ denotes the logistic activation function; W_{xi}^T, W_{xj}^T, W_{xo}^T and W_{xc}^T are the weight matrices of each of the four layers for their connection to the input vector x_t; W_{hi}^T, W_{hj}^T, W_{ho}^T and W_{hc}^T are the weight matrices of each of the four layers for their connection to the previous short-term state h_{t-1} ; b_i, b_f, b_o, b_c are the bias terms for each of the four layers; c_t is the long-term state at time t, and h_t is the output of the LSTM cell. In short, the input gate decides which parts of input at time t, should be added to the long-term state c_t; the forget gate stores the important part in c_t as long as it is needed, and output gate decides which parts of c_t should be read and output at a current time step. The decisions made by gate controllers are implemented through sigmoid activation functions whose outputs range from zero to one. Feeding output values close to zero to element-wise multiplication operation (⊗) makes the gates close and cell's states unchanged, while producing values close to one, makes them open and change the states.

The LSTM network which was developed for this study is illustrated in Fig. 4. As shown, it includes four layers: an input layer, an LSTM layer with a certain number of hidden cells, a dense layer and a reshape layer. The input layer includes $N \times L$ neurons, where N denotes the number of features and L denotes the number of temporal lags. The LSTM layer is used to grasp the internal representation of input data by capturing the deep temporal dependencies within the multivariate time series. The dense layer is responsible for forecasting the future values of the two targets based on the hidden state of the last LSTM cell in the first layer. Since each neuron in the dense layer is responsible for producing one output, the number of neurons in the dense layer is set to as twice as H steps ahead, corresponding to the forecasting requirements of the two targets. The reshape layer is finally used to transform the output layer to separated vectors for each target.

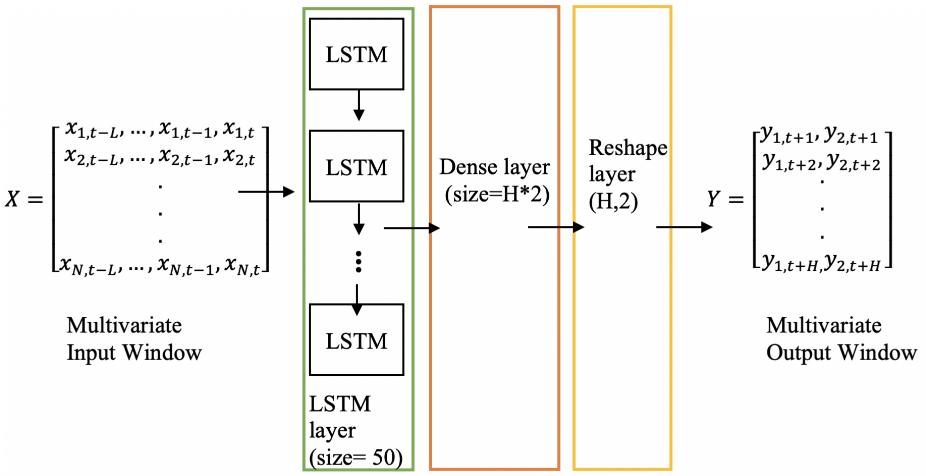

Figure 4: Structure of LSTM network for multivariate multi-step prediction

2.2.9 Gated Recurrent Unit (GRU) network

GRU, proposed by Cho et.al [49], is another version of LSTM with the same principles of processing long-term sequences but with a more compact structure. Compared to LSTM, it controls the information flow with fewer gates and parameters. GRU is thus trained faster and can be considered more effective in terms of simplified architecture. GRU has also successfully applied in both short-term residential load [50], and photovoltaic forecasting [51]. The architecture of a GRU cell is illustrated in Fig. 5.

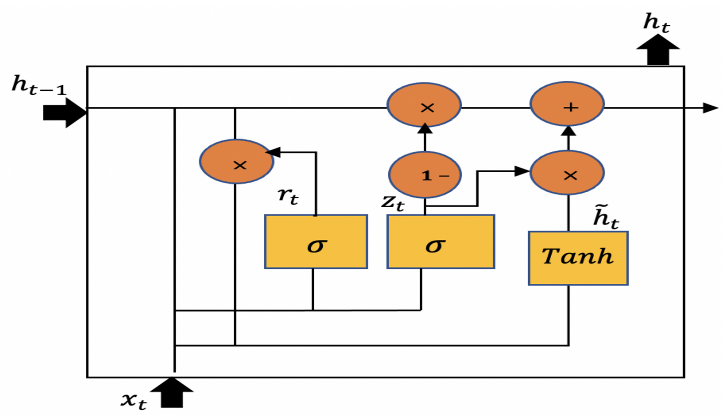

Figure 5: Structure of a basic GRU cell

In the GRU architecture, both state vectors are merged into a single vector h_t and a single gate both controls the input gate and forget gate. There is no output gate instead, the full state vector is output at every time step. However, there is a new gate controller which decides which part of the previous state will be available for the main layer. The equations from number 10 to 13 summarize the computation process of a GRU cell:

$$z_t = \sigma(W_{xz}^T.x_t + W_{hz}^T.h_{t-1} + b_z) \tag{10}$$

$$r_t = \sigma(W_{xr}^T.x_t + W_{hr}^T.h_{t-1} + .b_r) \tag{11}$$

$$\tilde{h_t} = \tanh\left(W_{xg}^T.x_t + W_{hg}^T.(r_t \otimes h_{t-1}) + b_h\right) \tag{12}$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h_t} \tag{13}$$

The GRU network architecture implemented for our study is similar to the one of LSTM network, which was illustrated in Fig. 4. The only difference is replacing LSTM cells by GRU cells in the recurrent layer.

2.2.10 Convolutional Neural Network (CNN)

CNNs are a branch of neural networks initially designed for such areas as speech recognition [52] and image classification [53]. They have been further applied in predicting energy time series data [54]. CNNs, compared to fully connected networks, are less complicated as they use fewer parameters to learn. They also do not require extensive feature engineering as they can automatically extract and generalize features from the input space.

Every convolutional network has three main components: (1) Convolution through one or multiple layers, where the features are extracted from input through filters, a non-linear transfer function and feature maps. Feature maps allow neurons in each convolution layer to be connected to neurons located within a small rectangle in the previous layer. This architecture enables the network to concentrate on low-level features in the first hidden layer and assemble them into higher-level features in the next hidden layers. (2) Pooling that reduces the dimensionality of feature maps while maintaining the relevant input information. (3) Fully connected layer that creates final non-linear combinations of features for making predictions by the network.

The architecture of the CNN network implemented for this study is depicted in Fig. 6. Six components are included: (1) an input layer that is the same as the one fed to LSTM and GRU networks. (2) The Convolutional layers that perform convolution operations on the multiple time series of the preceding layer with 32 and 16 filters including kernel size of two, followed by Relu layer; (3) a maximum pooling layer (as the most common type of pooling layer) that aggregates the inputs so that only the maximum value in each kernel passes through the next layer and the other inputs are discarded; (4) A flattening layer which transforms the 2-dimensional output to 1-d output; (5) a dense layer where neurons are connected to all the

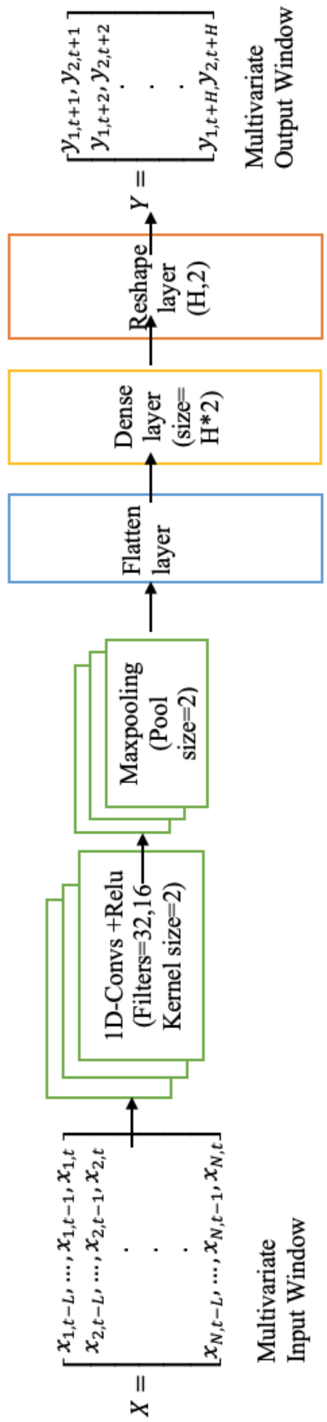

Figure 6: Structure of CNN network for multivariate multi-step prediction

neurons in the previous layer. The neurons in the dense layer are responsible for producing forecasts at multiple steps ahead and (6) the reshape layer used to transform the output shape to the desired shape.

2.2.11 Sequence-To-Sequence LSTM (Seq2Seq LSTM)

A Sequence-To-Sequence network (also called Encoder-Decoder) is a subclass of neural networks that can be used to map sequences to sequences. Encoder-Decoder networks have been proposed to implement machine translation systems in which the source language sentences are fed to the encoder and the destination language sentences are interpreted by the decoder.

A Seq2Seq network which utilizes LSTM cells in both encoder and decoder layers was first proposed by I.Sutskever et.al [55] for language translation. This network structure was further extended to time-series forecasting activities, especially targeting multi-step ahead forecasting.

The Seq2Seq architecture developed for our problem is adopted from this architecture, which performs multi-step forecasting of two targets based on multivariate input time series. As shown in Fig. 7, this network has two main components: one LSTM layer as the encoder and one LSTM layer as the decoder.

First, the input sequence is shown to the network one window at a time. Next, the LSTM encoder learns the relationship between time steps in the input. The output of the encoder shown as 'hidden states' layer in the architecture, is a vector v_t that contains the internal representation of the input series. The decoder converts this vector further into two target sequences as the multistep forward prediction values. The probability of each target sequence is then computed as Equation (14):

$$\begin{gathered} p(y_{j,t+1}, y_{j,t+2}, ..., y_{j,t+H|X_1,X_2,...,X_L}) = \\ \prod_{t=1}^{H} p(y_{j,t} \mid v_t, y_{j,1}, y_{j,2}, ..., y_{j,t-1}) \end{gathered} \tag{14}$$

Where y_j denotes the jth target variable for $j = (1, 2)$ and

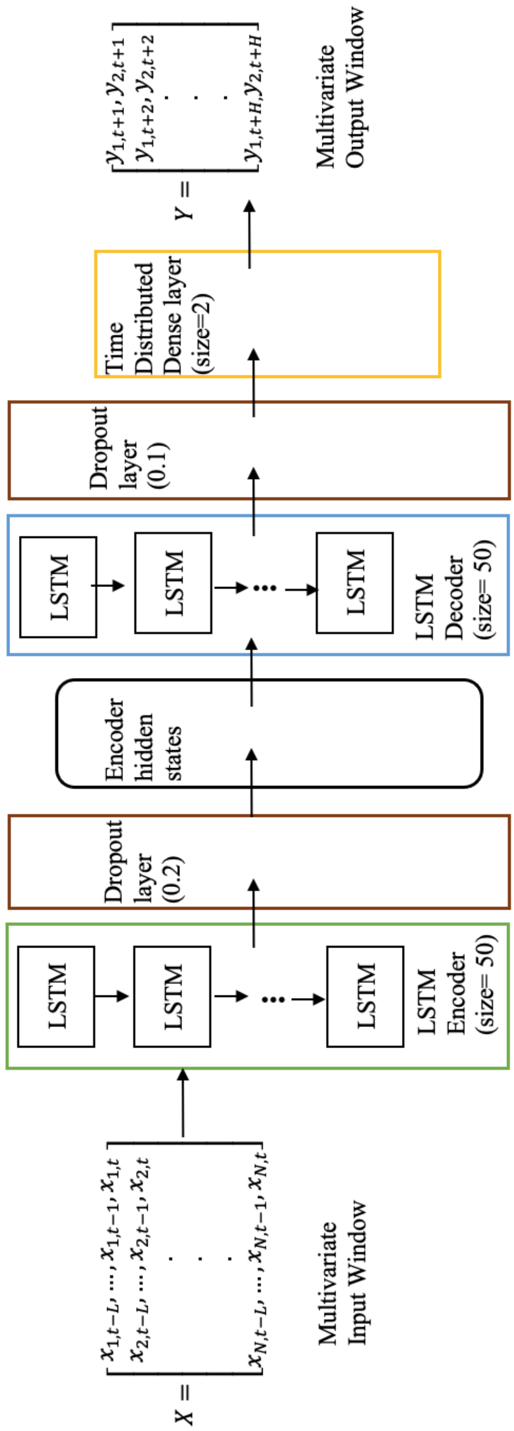

Figure 7: Structure of Seq2Seq LSTM network for multivariate multi-step prediction

$(X_1, X_2, \ldots, X_L)$ is a time series of multiple input series framed as a 2d-window; where X_i represents the ith column of this window and refers to the features values at ith time-step; and $(y_{j,t+1}, y_{j,t+2}, \ldots, y_{j,t+H})$ denote the forecasting values of H steps ahead of jth target value. Note that the number of temporal lags or window length L can be equal or different from the lookup size in the output sequences H.

2.3 Step Three

In this stage, we adopt an ensemble learning approach to create a strong forecasting model based on the algorithms evaluated in the previous step. When we combine the predictions of a group of predictors, the forecast accuracy is typically higher than that of the best individual predictor. The technique which utilizes the group of predictors is called ensemble learning. Ensemble learning can be performed in different ways.

One popular approach is called bagging, where predictors with the same training algorithms are trained on different random subsets of the training set. The sampling is performed with replacement and allows training instances to be sampled several times for the same predictor. After training of all predictors, the prediction is made for a new instance by simply aggregating the predictions of all estimators. As a result, the ensemble will have a lower variance than individual estimators. However, in the context of load forecasting, because of the inherent autocorrelation within the observations, the bagging method with a random sampling technique cannot be optimal.

Another common approach is boosting, where the predictors are sequentially trained, and each tries to correct its predecessor. The main idea of boosting is building a strong learner based on many weak learners. The major downside to sequential learning is that it can not be parallelized because each predictor's training takes place after training and assessment of the previous predictor.

A more advanced ensemble technique is called stacking. In this case, several predictors are trained on a subset of training data and make predictions on another subset of training data (called held-out set). A blender or meta learner then is trained on the forecasts of base predictors. In practice, the meta learner can be any learning algorithm such as Linear Regression or Decision Tree. A successful meta-learner effectively learns the optimal weights to combine the base learners and, as a result, produces more accurate predictions compared to the individual learners. Stacking is thus aimed at both reducing variance and improving the accuracy of forecasts. A detailed guide to ensemble learning is provided in [56].

In this work, we employ the stacking ensemble approach using the two most promising algorithms from Step Two. One algorithm with multiple variations is used to create the first-layer predictors (base learners) and one is used to merge the base learners' forecasts. We adopted and extended the development of base learners from [57] for processing multivariate input and output energy data. The training process and forecasting results of the ensemble model will be provided in Section 3 Subsection 8.

2.4 Dataset Description and Data partitioning

The original data was obtained from a publicly available dataset known as Ausgrid solar home electricity data [58]. It consists of half-hourly electricity consumption and generation of 300 Australian houses with rooftop solar systems from 2010 to 2013. Based on the recorded postal codes, two cities were recognized as the place of data collection: Newcastle and Sydney, each including 150 house profiles.

For this study, several aggregated profiles out of the original dataset were created to serve the purpose of energy forecasting at low aggregated levels, such as small household communities. More precisely, 150 individual profiles from each city were initially converted into three equal-sized groups, each including 50 members. The individual readings of energy consumption and generation from each group were summed up to create six aggregated load profiles in total over two cities. We considered each group as a small community and named them as A, B, C for Sydney and D, E, F for Newcastle. To provide smoother profiles and to be consistent with the frequency level of external datasets such as weather data, the 30-minute series was downsampled to hourly series using summation technique over one hour. Fig. 8 and Fig. 9 depict hourly electricity consumption and solar energy output of the six communities between 1 July 2010 and

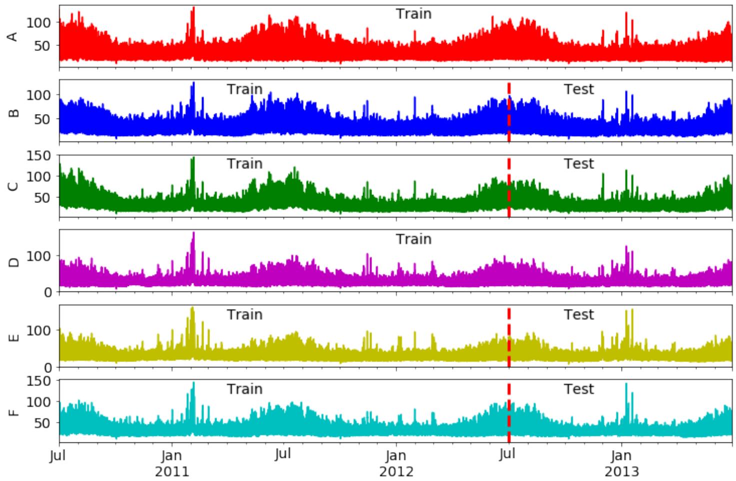

Figure 8: hourly total electricity consumption of six communities over three years

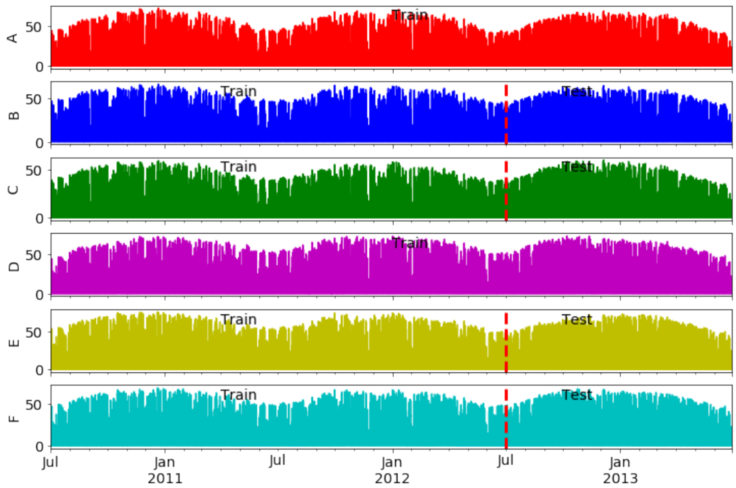

Figure 9: hourly total solar output of six communities over three years

30 June 2013 respectively.

As shown in the figures, the dataset was partitioned into different subsets with a total ratio of 77 % for training and 23 % for testing. The training subsets that were then concatenated together, covering the hourly energy data of all six communities over the first two periods (2010-11 and 2011-12) and the ones of A and D over the third period

(2012-13). The test subsets which were used for evaluation of the baseline models include energy consumption and generation of four communities over the third period. More precisely, the observations of Community B and C from Sydney, and the ones of Community E and F from Newcastle between 2012 and 2013 (The data on the right side of red dashed lines in Fig. 8 and Fig. 9). Each test subset was further divided into meta train and meta test sets with a ratio of 70 % and 30 % for building and evaluation of the ensemble method.

2.5 Data Exploration

Fig. 10 and Fig. 11 show the average values of hourly energy demand and supply in Kilowatt-hours by the six communities (A to F) over the year.

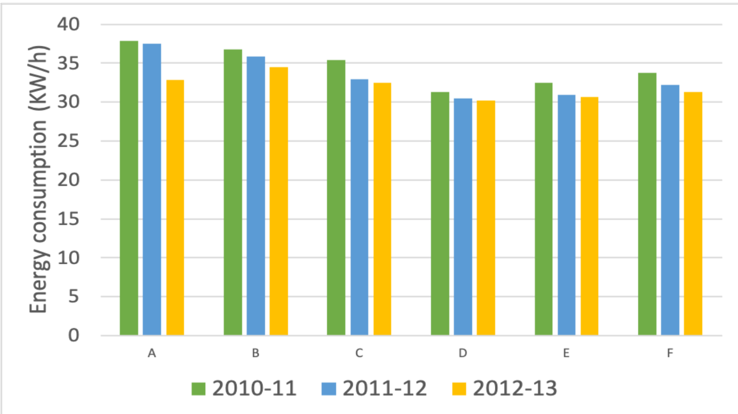

Figure 10: Mean hourly demand of six communities over the year

Overall, the households in Sydney (A, B and C) consumed more electricity than the consumers in Newcastle (D, E and F) in all three periods. Above 37 KW/h energy was spent in Community A between 2010 and 2011. This amount was the highest among other groups in different periods. Furthermore, both cities experienced a decreasing trend in energy consumption from 2010 to 2013 (Fig. 10). In terms of solar output, however, the people in Newcastle on average produced more energy in different periods; 13 KW/h as opposed to 10 KW/h (Fig. 11).

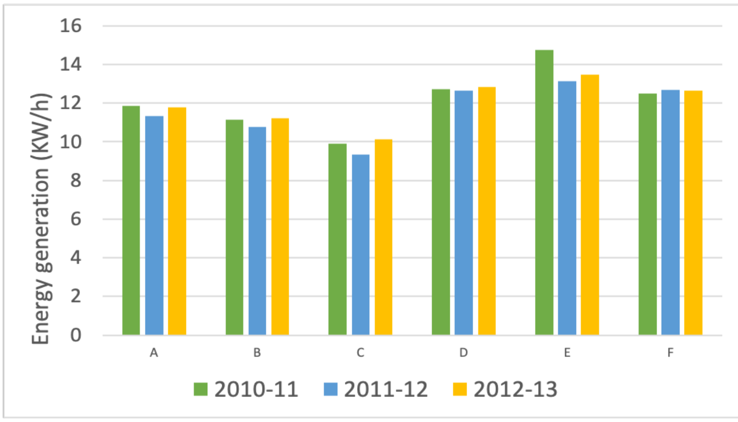

Figure 11: Mean hourly generation of six communities over the year

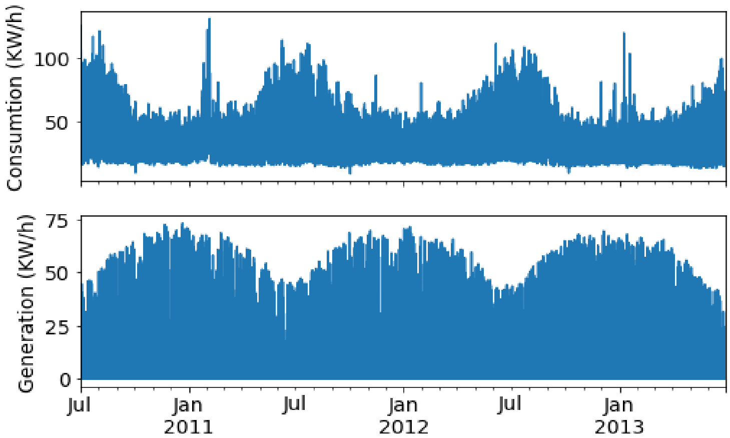

Figure 12: Hourly energy consumption and solar output of Community A over three years

Fig. 12. shows the consumption behaviour and production pattern of Community A as an example. There are up and down patterns in both consumption and generation profiles, as seen in the graphs, which are mainly replicated throughout the entire three-year cycle. We can also see that both profiles follow the time-of-year pattern in each period but in reversed directions. More precisely, during the cold months in Australia (May-September), with more usage of heating systems, the peak hourly electricity demands increases and

reaches around 100 to 120 KW/h, whereas the solar energy generation decreases at least by 20 percent (from 50 to 40 KW/h) during the same period with lack of sunshine. Similarly, the hourly electricity demand over non-cold periods (October-April) is reduced by half and reaches 50 KW/h in most intervals, whereas the energy generation grows continuously when in January reaches its highest amount; 60 KW/h.

The effect of time variables such as the month of year and day of the week on the load data is more visible in Fig. 13 and Fig. 14, where the mean aggregated loads of the houses (in Community A) vary by month and day with different degrees. According to Fig. 13, peaks in mean demand are seen around winter months (June, July, August) and troughs around March, April and October. The day-of-week pattern, however, is similar for all months except for July when the consumption reduces in the weekends. On the other hand, Fig. 14 indicates the maximum energy output levels in the summer months, such as December and January. Similar to the consumption plot, there is no regular pattern based on days of the week for solar energy generation.

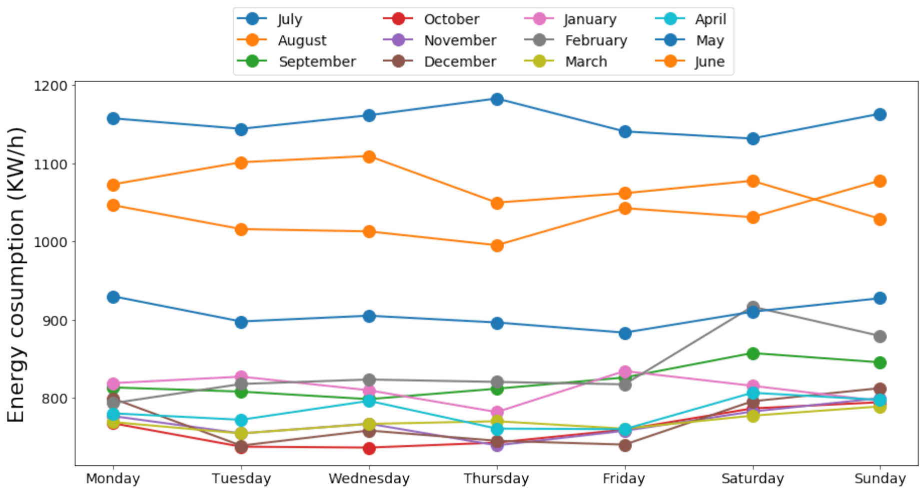

Figure 13: Average energy consumption by month and day

The analysis of time variables reveals the effect of outside air temperature and, most likely, other meteorological factors on household

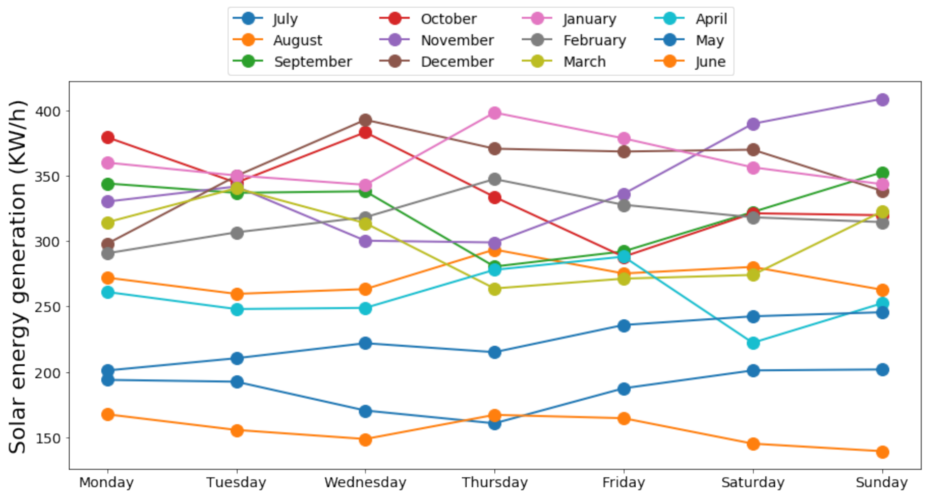

Figure 14: Average energy generation by month and day

consumption patterns and solar cell outputs. As stated in the introduction section, the effect of weather parameters in the issue of energy forecasting has also been demonstrated in the literature. Related meteorological data was collected for our analysis via the website of the Australian Government Bureau of Meteorology [59] and added to the aggregated load dataset.

Fig. 15 and Fig. 16 provide two examples of weather data analysis on the load profiles. Fig. 15 shows the demand during non-working hours plotted against outside temperature on both weekdays and weekends. The non-linear relationship indicates the importance of current air temperature in load demand prediction. The usage of air-conditioning systems for temperatures above 20 °C slightly lift the demand, while for lower temperatures around 10 °C, heating systems increase the demand considerably.

In Fig. 16, however, we can see the positive correlation between air temperature and solar output (during daytime). It also shows how various amounts of Global Horizontal Irradiance affect solar energy production. During hot days where the temperature is mostly above 20°, large concentrations of GHI are detected. Since the number of sunny days during the observing period is smaller, the overall amount of energy provided by solar panels is lower than that produced by

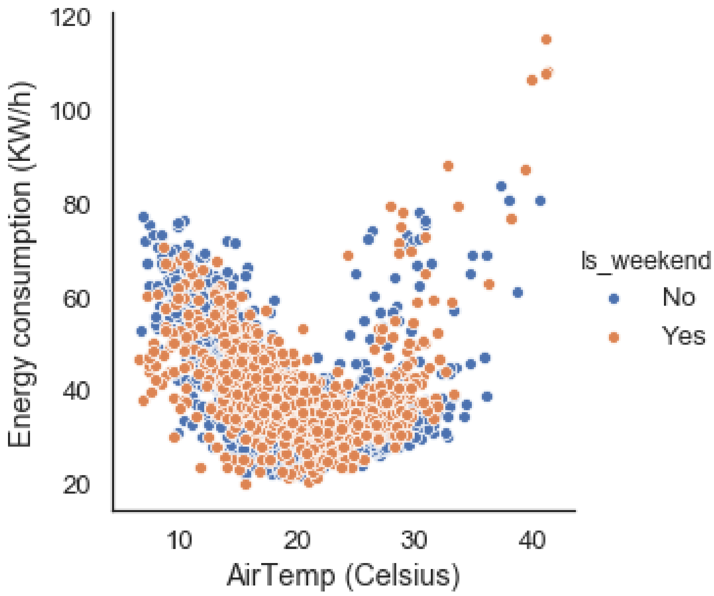

Figure 15: Aggregated load demand against air temperature

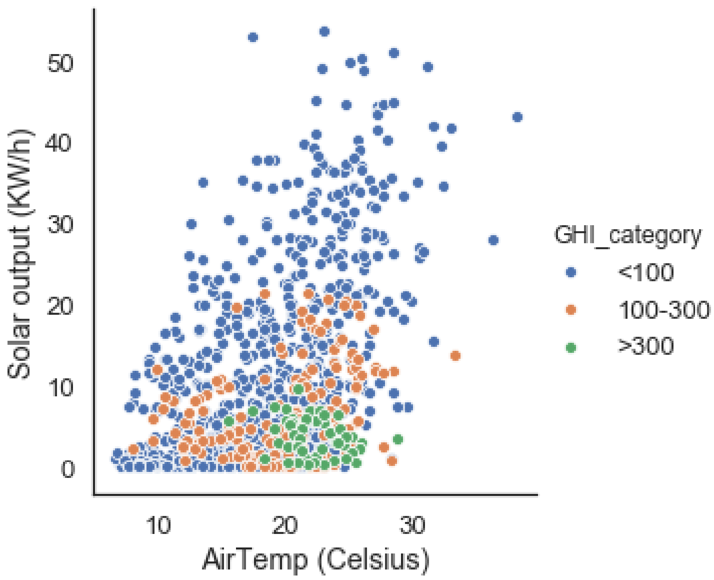

Figure 16: Aggregated solar output against air temperature

those days with a lower or moderate temperature.

In Fig. 17 and Fig. 18 we also looked at how demand and generation vary with the time of day and the weekend. Here, Hour 0 corresponds to 12 am to 1 am, Hour 1 corresponds to 1 am to 2 am, and so on. It can be seen that there are significant differences between daytime and night-time patterns of the two plots. As expected, the household peak consumption occurs during afternoon and evening while the generation peak happens around noon. Moreover, in both graphs, the weekend effect is relatively large, specifically during the daytime between 9.00 and 17.00.

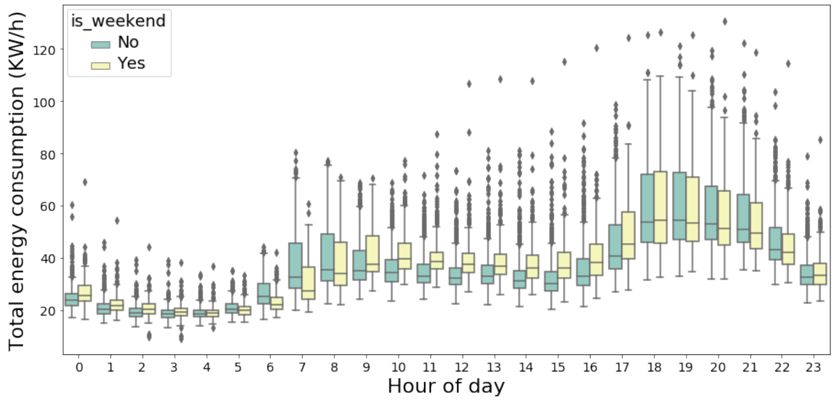

Figure 17: The hourly consumption distribution

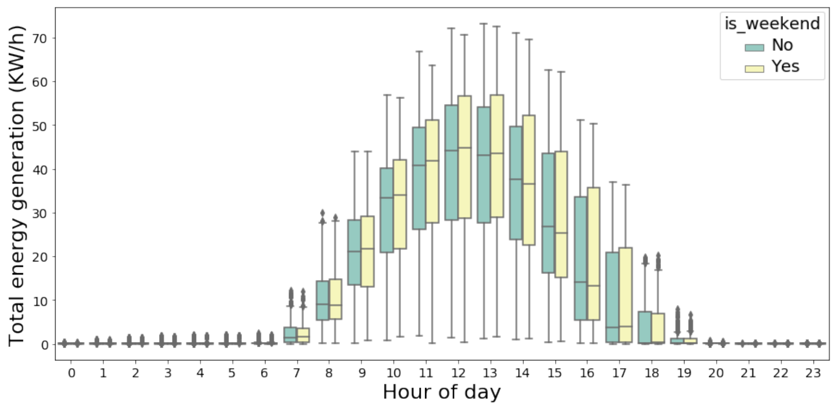

Figure 18: The hourly generation distribution

Future energy values can also be strongly correlated with the amount of energy spent or produced over previous steps in time. Fig. 19 illustrates the Partial Autocorrelation Correlation plots of the time series of hourly energy consumption and generation. The maximum lag step used for calculating PACs is set to $24 \times 7 = 168$ hours (i.e. one week) and the confidence interval is set to 85% shown by the light blue lines. We can see that in both graphs, PACs' absolute values mostly exceed the significance level up to 24 lags. This implies that the energy values at 24 previous hours are highly correlated with the energy values at the current hour.

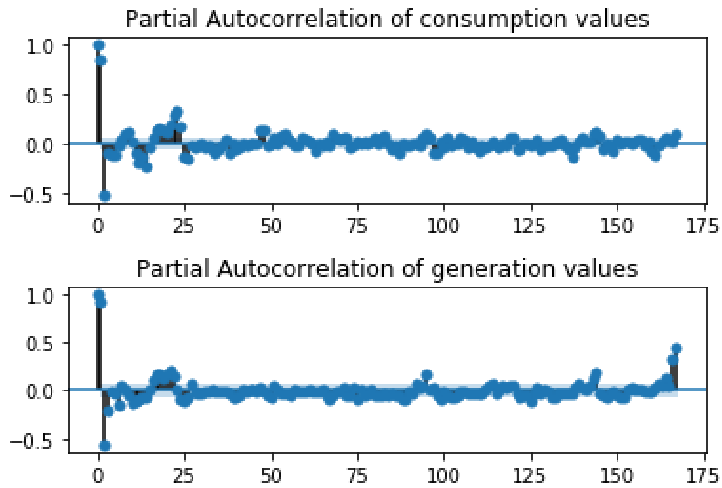

Figure 19: Partial auto-correlation plots

2.6 Data Cleaning and Feature Engineering

In the initial dataset, irregular profiles with a significant amount of incomplete records had been already removed. For our use, three additional cleaning activities were carried out on the dataset: One was to delete one consumer data from the Sydney dataset due to a large number of missed measurements for the three-year study period; The second was filling in missing values for the days with short gaps e.g. one to six hours. The third was to correct the record values indicating zero aggregated energy consumption, implying a situation that would be almost impossible in real-world scenarios. As mentioned in Section II, Part (2); we used EMA to replace the zero or empty measurements with valid values.

As mentioned in Section 2, Part 3; we created a candidate set of predictive features: historical energy load measurements, time and calendar variables along with meteorological features. Historical load variables refer to energy consumption and production of households at previous hours. The initial number of historical values is set to 24, as described in Section 3 and demonstrated in Fig. 19. Time and calendar variables include 'Hour of the day', 'Time of the year', 'Is weekend', and 'Is holiday'. The weather data include the following variables: Cloud Opacity %, Diffuse Horizontal Irradiance (DHI) in W/m2, Direct Normal Irradiance (DNI) in W/m2, Global Horizontal Irradiance (GHI) in W/m2, Solar Zenith angle (Zenith) in degree, Air Temperature in °C, Wind Speed in m/s, Wind Direction in °, Relative Humidity in %, and Precipitable Water in kg/m2.

Through feature transformation process, to create more meaningful inputs, the values of 'Hour of the day' and 'Time of the year' attributes were converted to integer values produced by *Sin* and *Cos* functions. This transformation represents the daily and yearly periodicity of load profiles in a more effective way. Additionally, the two categorical attributes representing weekends and holidays were formulated as binary features with One-hot encoding method.

Regarding the meteorological features,'Wind Direction' and 'Wind speed' were converted to more meaningful variables for the forecasting algorithms. Generally, wind direction in units of degrees is not considered as informative model input. For instance, 360° and 0° are close to each other and wrap around smoothly. If the wind is not blowing, then the direction should not matter. Therefore, we converted the 'Wind Direction' and 'Wind Speed' attributes to a wind vector with X and Y coordinates according to the following equations:

$$Wind\,(in\,radian) = WindDirection \times pi/180 \tag{15}$$

$$Wind\,X = WindSpeed \times \cos(Wind(in\,radian)) \tag{16}$$

$$Wind\,Y = WindSpeed \times \sin(Wind(in\,radian)) \tag{17}$$

Among the three features related to the sun radiation reaching the earth's surface, 'GHI' is of particular interest to photovoltaic installations and based on this formula $DNI \times \cos\theta + DHI$, it includes both 'DNI' and 'DHI'attributes; where θ refers to the angle of incidence of the beam. It is conceived that GHI can convey most information about its components to the model. Thus, to prevent the detrimental impact of feature redundancy on the performance of the predictive models, GHI remained in the feature set, and the other two were skipped.

As mentioned in Section 2, Part (5); we selected a subset of appropriate features from the candidate set through feature selection methods. The original feature set consists of 17 variables: 15 non-load variables, weather and time parameters along with two load variables, electricity consumption, and solar generation. Since we want to filter out non-load features from the original candidate set, we evaluate the feature selection methods using only non-load variables. We used one-year of training data with an hourly resolution to do the FS experiments.

The PCC method was designed to select the best n variables from the 15 predictors according to the correlation values higher than a threshold of +/- 0.3. For the RFE method, the Random Forest algorithm was utilized to identify m number of best attributes out of 15. In practice, the value of m is not known in advance. Therefore, in the first step, different values for the number of features were evaluated using the training data and a K-fold cross-validation technique for time series data with K equal to three. The temporal order of data is complied with in this cross-validation technique so that the model is tested on observations that have not been used as training data.

Fig. 20 and Fig. 21 demonstrate the distribution of mean absolute error (MAE) values for each configured number of input features (The MAE metric is further defined in Equation (18)). We can see that performance improves as the number of features increases. However, the reduction in MAE values continues until reaching a certain number in both graphs. By growing the input space, the median values (shown by orange lines in the boxplots) fluctuate and show no significant improvements. It implies mostly up to eight

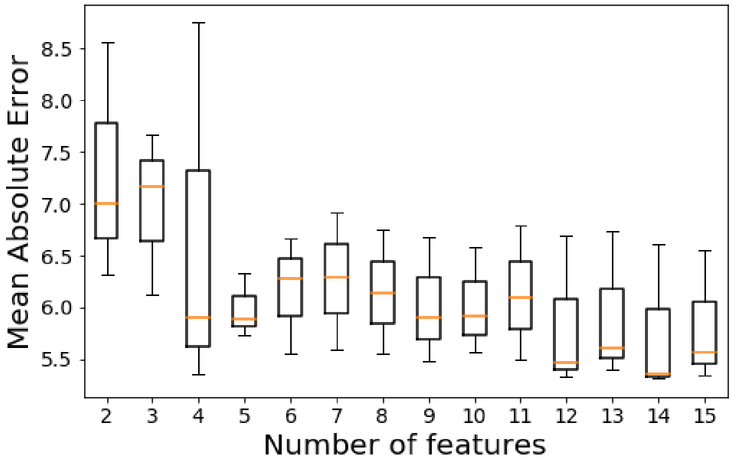

Figure 20: Error distribution by number of features for consumption estimation

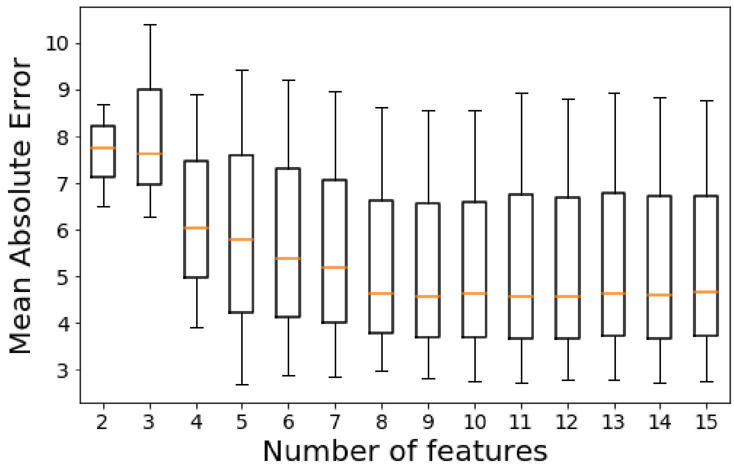

Figure 21: Error distribution by number of features for production estimation

or nine variables for both response variables can be relevant and influential.

In the next step, we used two conventional prediction algorithms known as support vector regression (SVR) and Random Forest (RF) to evaluate the prediction performance resulted by (1) the two FS techniques, (2) the combination of both and (3) with all features (without FS approach). Fig. 22 and Fig. 23 show the average MAE results over three folds regarding predicting the targets at one step ahead.

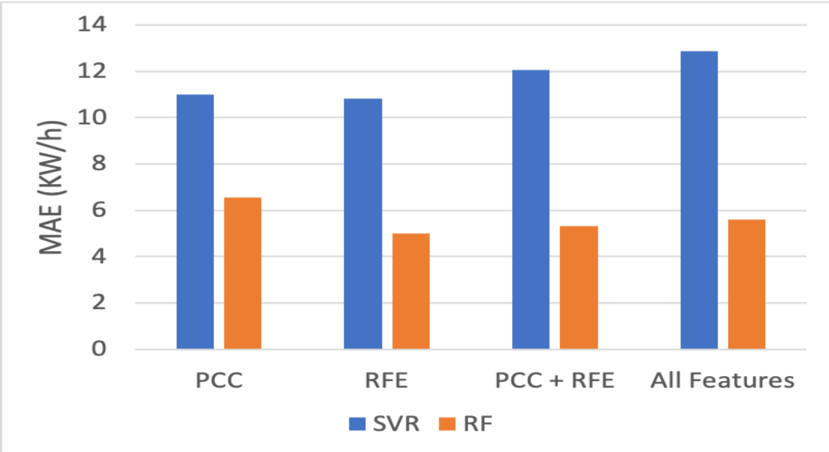

Figure 22: Average MAE using four feature sets for energy production

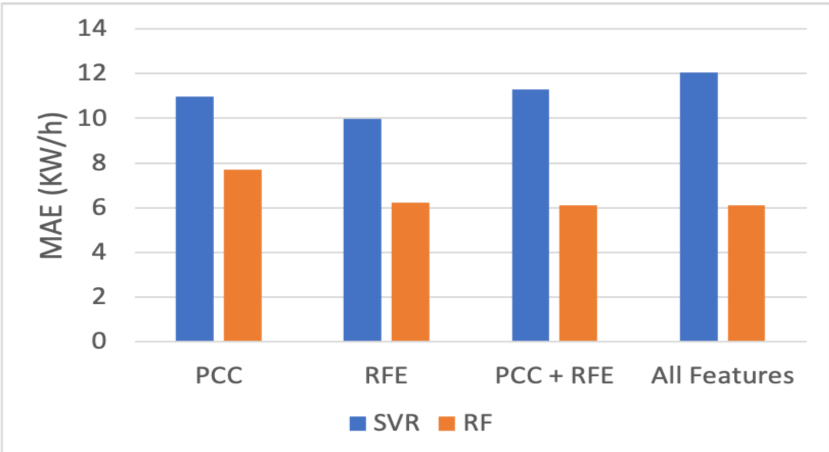

Figure 23: Average MAE using four feature sets for energy consumption

As shown, all feature selection methods show better MAE performance when using the RF model. The lowest MAE error for both SVR and RF is obtained by the RFE method for solar output prediction (Fig. 22). However, when it comes to consumption prediction, there is no unique variable set that produces the highest accuracy for both models (Fig. 23). As a result, the RFE-driven features which resulted in low MAE errors for both targets and by both predictive algorithms were chosen as the final feature set. Table 1 lists the final set; 12 non-lead variables out of 15.

Table 1: List of non-load features

Feature name	Value
Global Horizontal Irradiance	W/m2
Solar Zenith angle	Degree
Relative Humidity	Percentage
Precipitable Water	Kg/m2
Cloud Opacity	Percentage
Air Temperature	Celsius
Wind vector	$Wind\,Speed * Cos(Wind\,Direction)$
	$Wind\,Speed * Sin(Wind\,Direction)$
Time of day	$Sin(2 * \pi)/24 * hourofday$
	$Cos(2 * \pi)/24 * hourofday$
Time of year	$Sin(2 * \pi)/365 * 24 * hour\ of\ day$
	$Cos(2 * \pi)/365 * 24 * hour\ of\ day$

3 Forecasting Experiments and Results

In this section, we describe the experiments conducted in Step Two and Step Three and analyze the forecasting results.

3.1 Implementation Environment

All models were implemented using the Scikit-learn open-source machine learning library and the Keras framework for deep learning. The experimental hardware environment is based on a 3.1 GHz Intel (R) Core i5 CPU and 16 GB of memory.

3.2 Data Scaling and Input Requirements

Since both conventional and deep neural networks are sensitive to the input scale and are more efficiently trained with normalized data, the data was normalized to the range [0, 1] by applying the Min-Max function. All predictive algorithms except for Persistence and ARIMA used the normalized input comprising the final set of variables; 12 non-load features presented in Table 1 along with one or two load (energy) features. ARIMA and Persistence only used historical energy data to predict future values of the two targets.

For the Ridge regression, SVR, AdaBoost, GBRT and BPNN, each input window explained in Part (6) of Section II, was transformed into a flattened format of $L*(N-1)$, where L is equal to 24 lags and $N-1$ is equal to 13 as the total number of variables (12 non-load variables selected through FS process and 1 load variable). As mentioned in Section II, Part (7), reducing one load variable from the input with the size of N indicates that the model that predicts future values of one target removes the values of other target from its input.

All deep models, on the other hand, are fed with 2-D input windows containing all $N = 14$ variables. Since they are capable of producing two outputs at the same time they are fed with both load varibles in addition of other non-load features. As indicated in Fig. 19, the length of input windows L was initialized as 24 for all the experimental models in Step Two. At the same time, this value was tuned as the hyperparameter of the final forecasting algorithm in Step Three. The forecast horizon H as the length of output windows was set to 24 for all experiments.

3.3 Multi-step and Multi-variate Forecasting Strategy

To predict the observations at multiple time steps, we applied two strategies depending on the training model:

(1) Direct multi-step forecast strategy for the models which naturally do not support multi-output regression including Ridge regression, SVR, AdaBoost, and GBRT; For each forecast time step, one model is developed (e.g. 24 models for 24 steps ahead).

Furthermore, since the given models are not adopted for predicting multivariate time series, a separate model was trained and evaluated for each prediction target; one for energy consumption estimation and another for energy generation prediction.

(2) Multiple output strategy for the models capable of performing multi-output regression including BPNN, CNN, LSTM and Seq2Seq LSTM. This strategy involves the development of one model capable of predicting the entire forecast sequence at once. Unlike BPNN that considers the inputs as independent variables, the rest of ANN-based models, due to their learning procedure, can learn the dependency structure between inputs and outputs as well as between outputs. Therefore, they become more complex and are expected to perform better with sufficient training data. Moreover, as the deep models can produce multivariate outputs, individual deep models were trained to produce two targets time series at once.

3.4 Parameter Settings

We mostly used the default values considered in the Scikit-learn library for the conventional ML models in terms of model parameter configuration. However, we modified the values of some of the parameters specified in Table 2, according to the input and output specifications, as well as the complexity of the prediction task.

For the recurrent deep models, the same parameters of neural network architecture configuration (Table 2) were used; for the LSTM and GRU 50 neural unit in one hidden layer and for the Seq2Seq LSTM, the same number of neurons was used in both encoder and decoder each including one LSTM layer. For BPNN, to have a fair comparison with deep models, we increased the number of hidden layers to two with 60 and 30 neural units besides increasing the number of training epochs from 80 to 100. For all ANN-based models, 'Relu' [60] was applied as the activation function of hidden layers, mean square error (MSE) was used as the loss function, and 'Adam' function [61] was set as the model optimizer. The batch size sets

Table 2: Parameter settings of experimental models

Model	Parameter	Value
ARIMA	p,q,r	2,2,0
Ridge Regression	Regularization parameter(Alpha)	1.0
SVR	C, Gamma, Epsilon, Kernel fucntion	100, 0.1, 0.1, RBF
AdaBoost,GBRT	Maximum depth of the tree, number of trees, Min samples split , Min sample leaf	3, 150, 2, 2
BPNN	Default hidden layer, Units in hidden layer, Training epochs,	2, (60,30), 100, 32,
	Batch size, Loss function, Optimizer, Activation function of hidden layer	mse, Adam, Relu
CNN	Conv-layer, Kernel size, Filter size, Training epochs,	2, 2, 32, 80, 32
	Batch size, Loss function, Optimizer	mse, Adam
LSTM,GRU	Default hidden layer, Units in hidden layer, Training epochs	1, 50, 80, 32, 0.1, mse
	Batch size, Drop out, Loss function, Optimizer, Activation function of hidden layer	Adam, Relu
Seq2Seq LSTM	Default hidden layer (Encoder), Default hidden layer (Decoder),	1, 1, 50, 80, 32, 0.1
	Units in hidden layers, Training epochs, Batch size, Drop out,	mse, Adam, Relu
	Loss function, Optimizer, Activation function of hidden layers	

to 32, learning rate sets to 0.001 and the drop out rate sets to 0.1 (excluding BPNN and CNN which did not use dropout layer).

3.5 Evaluation Metrics

To evaluate the prediction results of the trained algorithms, two commonly used error metrics in time series forecasting are used; Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Lower values of error metrics indicate a more accurate prediction. The MAE measures the difference between predicted and real values on average and ignores whether the prediction values are greater than or smaller than the actual values. The RMSE, in contrast, penalizes large errors before averaging them by computing the square error. These two metrics are defined as follows:

$$MAE = \frac{\sum_{t=1}^{N} |\hat{y}_i - y_i|}{N} \tag{18}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (\hat{y}_i - y_i)^2}{N}} \tag{19}$$

Where y_i and $\hat{y}_i$ denote actual and predicted output at time step t and N denotes the number of sample observations in the testing period.

Since focus of this work is on multi-step forecasting, we also added another metric and named it as SDE to evaluate the consistency of the errors throughout the whole forecast horizon. The SDE metric computes the standard deviation of mean error values in each time step throughout the forecast horizon and is defined as follows:

$$SDE = \sqrt{\frac{\sum(\overline{E_i} - \mu)}{N}} \tag{20}$$

Where $\overline{E_i}$ denotes the average error values over timestep i in the forecast horizon, μ denotes the mean of $\overline{E_i}$ and N denotes the total number of steps in the forecast horizon i.e 24. The lower value of SDE indicates lower variations and consequently, more stability at multi-step ahead forecasting.

3.6 Experiments in Step Two

To build and evaluate all models in Step Two, a sub-sample of training data (60%) was considered sufficient equal to nine yearly periods of hourly observations from different communities. A cross-validation methodology known as blocked cross-validation was applied instead of a standard train-test split to avoid overfitting and measuring each model's performance more robustly.

In this technique, which is designed for time-series data, the sample training set is split into n non-overlapping subsets. At the first iteration, the first subset is further divided into two-folds on the condition that the validation set is always ahead of the training set. For the next iteration, the next subset is again divided into two folds and the iterations continue until n times. As a result, the temporal dependencies between observations are preserved during testing and also no leakage from future data is introduced to the model. In each iteration, the model will not observe and memorize patterns from an iteration to the next. In this study, the number of iterations was set to three and the division rates for each subset are set to 80% for train fold and 20% for validation fold. This split method is depicted in Fig. 24 for more clarification.

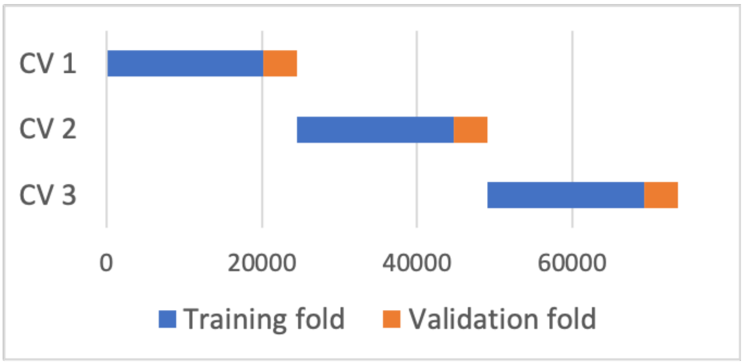

Figure 24: Blocked cross-validation with three splits

The vertical axis refers to the number of cross-validation iterations whereas the horizontal axis represents the size of training data on an hourly basis. The training folds are depicted in blue, and the folds used for validation are depicted in orange. The dataset has not shuffled, and the chronological order is preserved along the horizontal axis.

3.7 Results Analysis of Step Two

Table 3 and Table 4 provide multi-step a head forecasting results of different models using the cross-validation technique for two prediction targets. In each table, the three columns on the left report average RMSE errors in KW/h over 24 time steps per validation fold. The last column on the right provides the average RMSE errors over the three folds with standard deviation values.

Table 3: Average RMSE using the blocked cross-validation technique for energy consumption prediction

Model	Fold 1	Fold 2	Fold 3	Over three folds
Persistence	24.09	30.29	34.86	29.75 +/-4.41
ARIMA	16.9	19.09	18.94	18 +/- 0.89
Ridge Regression	6.60	6.42	6.84	6.62 +/- 0.17
SVR	7.97	8.06	9.13	8.93 +/- 0.52
AdaBoost	11.02	10.43	11.13	10.86 +/ -0.30
GBRT	5.80	5.67	6.32	**5.93** +/- 0.28
CNN	5.88	6.30	6.30	6.16 +/- 0.19
LSTM	5.78	5.85	6.43	6.02 +/- 0.52
GRU	5.76	6.08	6.41	6.09 +/- 0.26
Seq2Seq LSTM	5.92	5.85	6.30	**5.91** +/- 0.29

The experimental results show that different learning algorithms outperform the Persistence technique showing at least 40% to at most 80% improvement in prediction accuracy. We can see the Seq2Seq LSTM followed by GBRT outperform the other techniques in terms of energy consumption estimation. Regarding energy production forecasting, the lowest prediction error is achieved by the GBRT and GRU. Further analysis reveals that RMSE of deep neural network models (e.g. Seq2Seq LSTM, LSTM, CNN) are similar and lower than

Table 4: Average RMSE using the blocked cross-validation technique for energy production prediction

Model	Fold 1	Fold 2	Fold 3	Over three folds
Persistence	34.99	30.56	34.58	33.8 +/- 1.99
ARIMA	12.03	11.55	12.4	12+/-0.34
Ridge Regression	6.43	5.66	6.72	6.27+/-0.44
SVR	7.68	4.44	7.15	6.42 +/- 1.42
AdaBoost	9.41	7.72	10.75	9.29 +/ -1.23
GBRT	5.77	4.88	5.97	**5.41** +/- 0.65
CNN	6.24	5.01	5.99	5.74 +/ -0.52
LSTM	6.71	4.20	5.87	5.59 +/- 0.93
GRU	6.46	4.12	5.76	**5.45** +/- 0.98
Seq2Seq LSTM	6.86	3.90	5.85	**5.54** +/ -0.29

that of shallow neural network (BPNN) as well as most conventional learning models (e.g. ARIMA, Ridge Regression, and SVR). Among the deep models, the Seq2Seq LSTM yields low average RMSE errors (less than 5 KW/h) with a low standard deviation (0.29) for both demand and PV output forecasting indicating more accurate and robust performance against other deep models.

To further investigate each method's effectiveness in terms of computational cost, we have computed the average training time over three folds. Each training fold covers around two years and three months of hourly data and occupies about 2.6 Mega byte of the system memory. Note that for the models that do not support multivariate regression, the training time is multiplied by two to represent the required training time to forecast the two energy targets. The persistence model is ignored in this evaluation since it does not pass any training phase. Table 5 presents the results.

ARIMA and Adaboost not only produce inaccurate hourly forecasts but also require long training times. Ridge regression followed by BPNN is substantially faster among all methods. SVR compared to deep models needs higher training time. While GBRT algorithms appear to have good precision (according to Table 4), relative to

Table 5: Total training time for 24-steps ahead prediction of two targets

Model	Total training time (Seconds)
ARIMA	1100 *2 = 2200
Ridge Regression	3*2 = 6
SVR	171 *2 = 342
AdaBoost	1480 *2 = 2960
GBRT	670 *2 = 1340
BPNN	38 *2 = 76
CNN	90
LSTM	420
GRU	400
Seq2Seq LSTM	830

other methods, they are slower. Among deep neural networks, CNNs are the fastest, followed by GRU and LSTM networks.

In conclusion, among the candidate techniques, Seq2Seq LSTM and GBRT were chosen as the most promising models for building the ensemble model in the next step. Despite being slower than other algorithms, these models have shown higher forecasting accuracy for both predictive targets. To accelerate the training procedure, GPU-based computing can be adopted in real-world scenarios.

3.8 Experiments of Step Three

3.8.1 Ensemble Setting

Having access to a large training set (14 years of hourly data from 6 communities) inspired us to use the deep Seq2Seq LSTM models as the first-layer predictors or base learners and the GBRT algorithm as the meta learner to capture non-linear relations between base predictions. To build an ensemble model with large diversity, multiple Seq2Seq-LSTM networks were developed and parameterized differently. The parameters that contributed to the ensemble's diversity include learning rate, number of hidden layers in Encoder, number of hidden neurons in Encoder and Decoder, type of layer in Encoder, and the length of the input sequence (windows) W.

For evaluation, we chose $W = [24, 24 * 2, 24 * 3]$ as the length of input windows, and for each of the other parameters, a set of values were considered from a candidate set V. As a result, we ended up with $W * V$ Seq2Seq LSTM models for each given parameter. For our experiments, the V vector for each given parameter is set to values as the following: learning rate : {0.01,0.001}, the number of hidden layers in Encoder : {1,2}, the number of hidden neurons in Encoder and Decoder : {60,90} and the type of Encoder layer : {LSTM, BILSTM}. As default values, we chose ADAM as an optimizer, mean squared error as the loss function, batch size of 64 and 20 iterations as the number of training epochs per network.

For the stacking purpose, the forecasts from the first-layer predictors are treated as input features for the GBRT. As we want to predict future values of two variables (energy consumption and generation), and the GBRT does not support multivariate output regression, one GBRT model was trained per target based on the corresponding predictions from the first layer. The performance of meta learners was then evaluated against real observations available to us within the test set.

All four test sets (mentioned in Section III, Subsection A) were used to evaluate and demonstrate the prediction capabilities of the ensemble approach. Each test set representing a community load for one year was divided into two subsets with 70% and 30% ratios known as a meta train set and a meta test set. The meta train set was used to build a training set for the meta learner (GBRT) so that all trained LSTM base learners in the first step were tested on this set.

The meta train set was used to build a training set for the meta learner (GBRT) so that all trained LSTM base learners in the first step were tested on this set. The Meta test set was then used to evaluate the performance of the meta learner against three other techniques:

- The Seq2Seq LSTM network that yields the lowest prediction error among the individual learners (called the best learner).

- The ensemble that computes the average of the base learners' predictions for each step in the forecasting horizon.
- Another stacked ensemble of Seq2Seq LSTM networks which applies Ridge regression as the meta learner to capture linear relations between the first layer's forecasts.

It is worth mention that none of the meta test sets has been seen by any learning algorithm during training.

3.8.2 Result analysis of Step Three

Fig. 25 and Fig. 26 provide example comparisons between the actual (Ground-truth) and predicted energy load curves using various approaches in four test communities. We can see that in all graphs, on some test steps, the forecast precision decreases due to irregular fluctuations in hourly load, especially for load demand prediction. Nevertheless, in most cases, all models, specifically the two ensemble approaches, could effectively follow the usage and generation patterns for both forecasting targets.

Table 6 and Table 7 summarise the forecasting results of different methods. For each test set, the average MAE and RMSE are calculated for three horizons: short-term; from 1 to 8 hours ahead, medium-term; from 8 to 16 hours ahead, and long-term; from 16 to 24 hours ahead.

It is observed that the stacked ensemble with GBRT and Ridge algorithms have the lowest average MAE and RMSE overall forecast horizons and across all test sets for both energy targets. The prediction accuracy of electricity consumption and PV power output can be effectively improved by using ensembles of Seq2Seq LSTM networks.

After using a conventional ML algorithm on top of them, the forecast accuracy can be further improved since more diverse first-layers' forecasts can provide more relevant information for model training. This implies that the ensembles with GBRT and Ridge regression can model regular and irregular patterns of future energy values more effectively.

In contrast, the performance of the averaged ensemble compared to the other three algorithms is not highly accurate and stable across

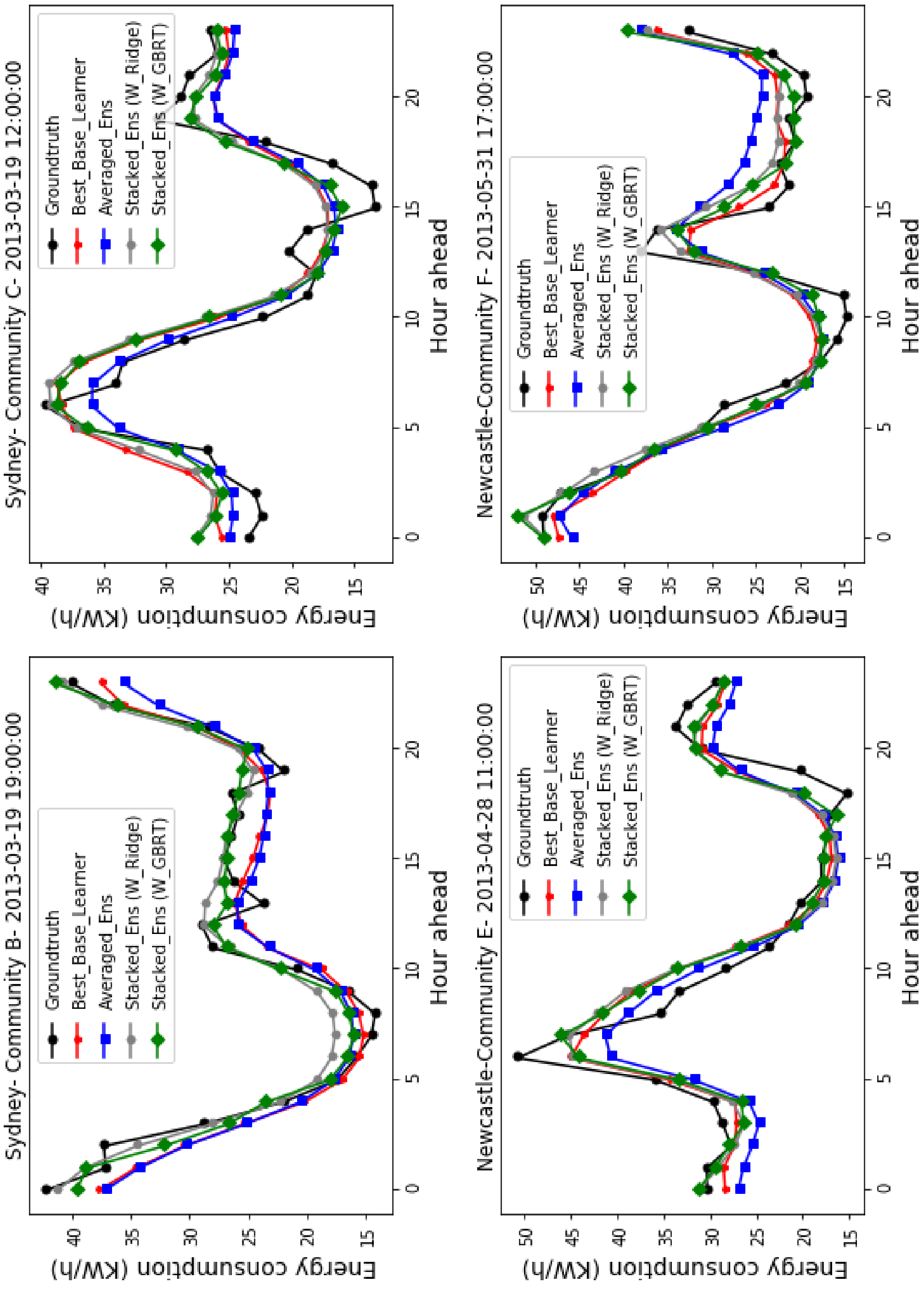

Figure 25: Comparison of 24-h ahead load consumption forecasting results

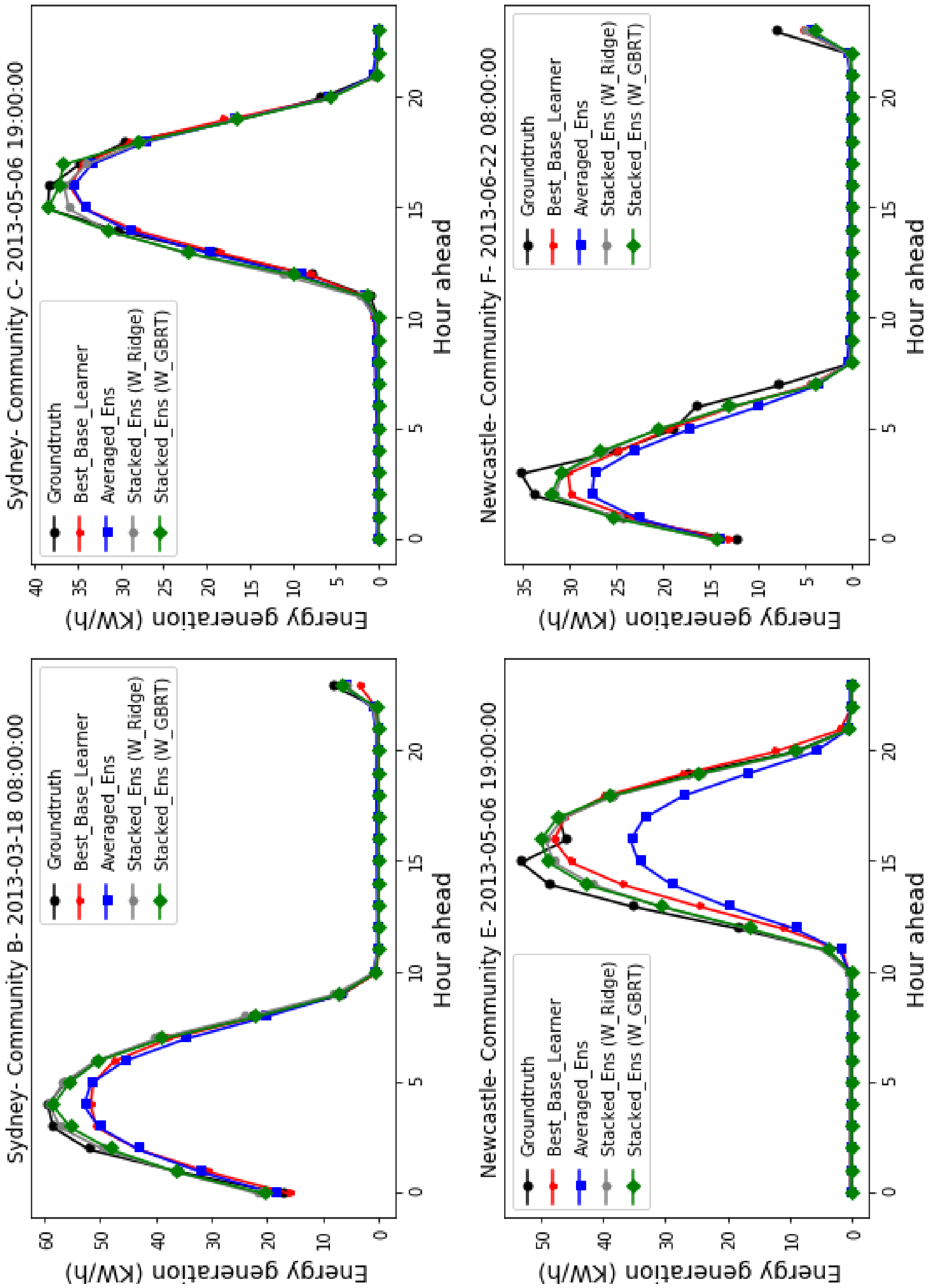

Figure 26: Comparison of 24-h ahead energy generation forecasting results

Table 6: Average error metrics over multiple steps ahead for energy consumption prediction

Model/Horizon	Test set 1 (1-8)	Test set 1 (8-16)	Test set 1 (16-24)	Test set 2 (1-8)	Test set 2 (8-16)	Test set 2 (16-24)	Test set 3 (1-8)	Test set 3 (8-16)	Test set 3 (16-24)	Test set 4 (1-8)	Test set 4 (8-16)	Test set 4 (16-24)
Average MAE												
Best base learner (Seq2Seq LSTM)	4.15	4.09	4.10	3.77	3.85	3.87	3.22	3.27	3.23	3.70	3.85	3.74
Averaged ensemble	4.06	4.31	4.33	3.69	3.90	3.92	4.24	4.17	4.20	3.82	3.90	3.88
Stacked ensemble (Seq2Seq LSTM + Ridge)	3.63	3.80	3.76	3.49	3.66	3.65	**3.01**	3.09	3.13	3.49	3.60	3.61
Stacked ensemble (Seq2Seq LSTM + GBRT)	**3.59**	**3.73**	**3.72**	**3.42**	**3.57**	**3.59**	3.03	**3.01**	**3.05**	**3.31**	**3.34**	**3.28**
Average RMSE												
Best base Learner (Seq2Seq LSTM)	5.61	5.59	5.72	5.06	5.31	5.32	4.18	4.26	4.23	4.81	4.98	4.89
Averaged Ensemble	5.58	5.96	6.09	5.10	5.45	5.60	5.40	5.29	5.33	4.97	5.05	5.07
Stacked Ensemble (Seq2Seq LSTM + Ridge)	4.83	5.06	**5.04**	4.65	4.90	**4.92**	**3.97**	4.03	4.12	4.55	4.65	4.68
Stacked Ensemble (Seq2Seq LSTM + GBRT)	**4.81**	**5.01**	5.05	**4.61**	**4.85**	4.94	4.05	**3.97**	**3.96**	**4.37**	**4.40**	**4.38**

Table 7: Average error metrics over multiple steps ahead for energy generation prediction

Model/Horizon	Test set 1 (1-8)	Test set 1 (8-16)	Test set 1 (16-24)	Test set 2 (1-8)	Test set 2 (8-16)	Test set 2 (16-24)	Test set 3 (1-8)	Test set 3 (8-16)	Test set 3 (16-24)	Test set 4 (1-8)	Test set 4 (8-16)	Test set 4 (16-24)
Average MAE												
Best base learner (Seq2Seq LSTM)	1.43	1.67	1.88	1.01	1.15	1.34	1.41	1.60	1.74	1.24	1.42	1.56
Averaged ensemble	1.46	1.79	2.13	1.03	1.30	1.60	4.05	4.15	4.38	1.42	1.70	2.11
Stacked ensemble (Seq2Seq LSTM + Ridge)	1.13	1.33	1.55	0.88	1.05	**1.20**	1.19	1.31	1.55	1.02	1.15	1.35
Stacked ensemble (Seq2Seq LSTM + GBRT)	**1.04**	**1.29**	**1.52**	**0.87**	**1.04**	1.24	**1.12**	**1.26**	**1.46**	**0.98**	**1.13**	**1.33**
Average RMSE												
Best base Learner (Seq2Seq LSTM)	2.65	3.19	3.74	2.01	2.35	2.77	2.63	3.09	3.36	2.36	2.75	3.05
Averaged Ensemble	2.74	3.33	3.95	1.98	2.47	2.99	7.50	7.60	8.04	2.62	3.09	3.94
Stacked Ensemble (Seq2Seq LSTM + Ridge)	2.33	**2.83**	**3.32**	**1.83**	2.19	**2.55**	2.24	**2.57**	3.09	2.02	**2.33**	**2.79**
Stacked Ensemble (Seq2Seq LSTM + GBRT)	**2.28**	2.86	3.39	1.87	2.26	2.69	**2.22**	2.59	**3.06**	**2.01**	**2.33**	2.82

different test sets. For instance, on Test set 3, the predictive errors for solar output prediction are far greater than those obtained by other algorithms. The explanation is that unlike stacked ensembles, the forecasting performance of the Averaged model is equally determined by all contributors' forecasts. With even one poor base estimator, performance will degrade significantly.

Fig. 27 and Fig. 28 illustrate the prediction performance of each algorithm on average from 1 to 24 steps ahead per test set for the two targets. They confirm the results of previous experiments.

Regarding energy consummation prediction, the Averaged ensemble produces higher MAE (4.27, 3.86, 4.19, and 3.88) compared to the best base learner (4,14, 3.84, 3.25, and 3.77). In terms of RMSE, the forecast error of the Averaged model (5.93, 5.41, 5.32, and 5.05) is also higher in comparison with the best LSTM network in the ensemble (5.68, 5.25, 4.23, and 4.91). The proposed forecast framework (Seq2Seq LSTM + GBRT) produces an average MAE of 3.39 and RMSE of 4.55 over four test sets, which are lower than those of all other models.

Similar forecast accuracy is recorded for the PV power output of the communities. The best LSTM network gives more accurate results than the Averaged ensemble forecast, but lower compared to the stacked models. The stacked ensemble with GBRT slightly outperforms the Ridge-based stacked ensemble with 2.4% reduction in MAE score on average across the test sets. However, it significantly produces more accurate forecasts than the best learner and averaged ensemble, showing on average 17% and 47% reduction in MAE as well as 10% and 37% reduction in RMSE.

To evaluate the consistency of errors throughout 24 time steps, we calculated the average MAE values of the prediction results at each time step for different models. Fig. 29 and Fig. 30 illustrate the results. We can see that the forecasting errors of all models fluctuate smoothly along the forecast horizon up to 10 steps and then increase with varying degrees for different models.

Figure 27: Average error metrics for energy consumption prediction from 1 to 24 steps ahead

Figure 28: Average error metrics for energy generation prediction from 1 to 24 steps ahead

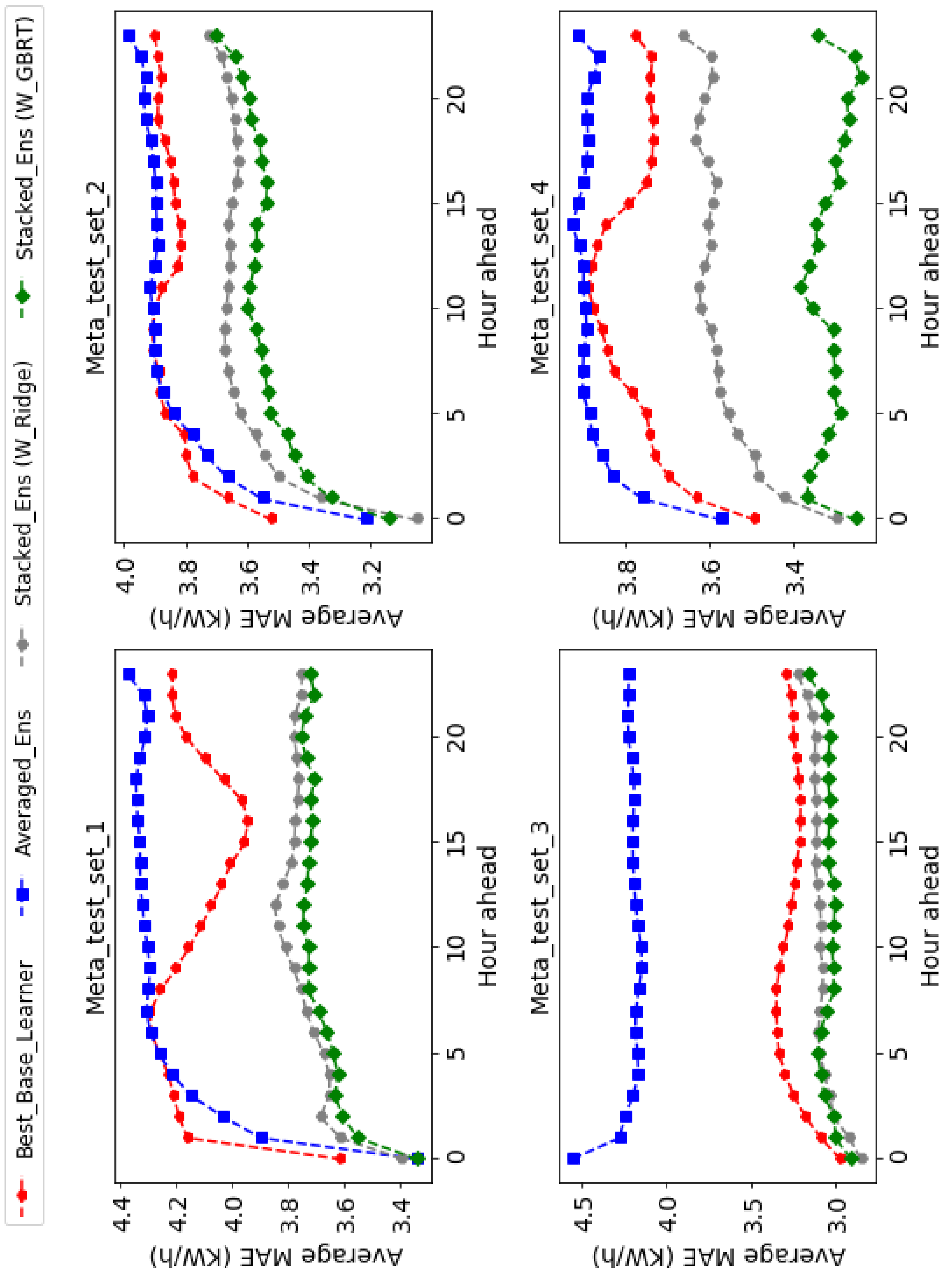

Figure 29: Average MAE for energy consumption prediction over each time step on four Meta test sets

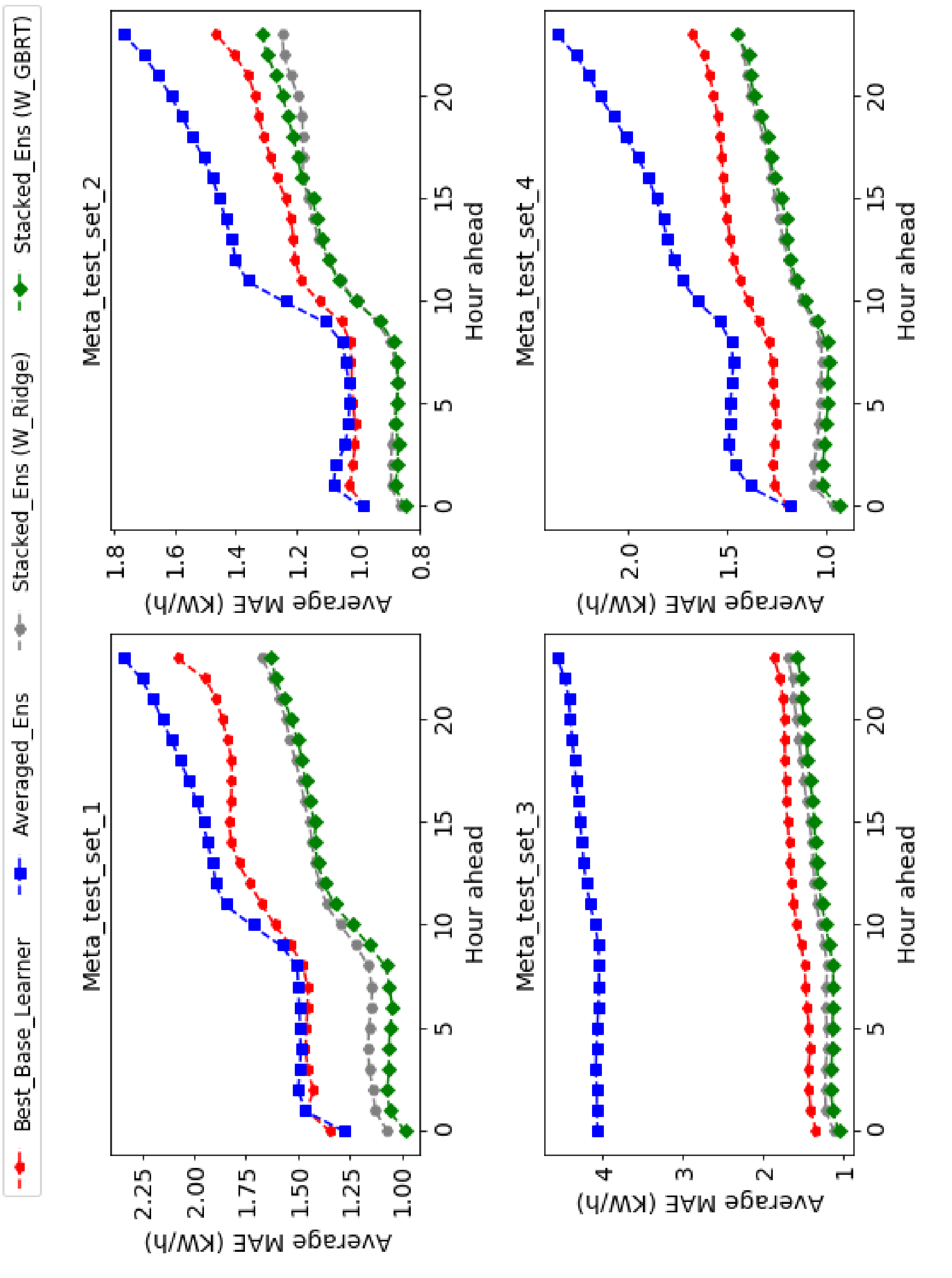

Figure 30: Average MAE for energy production prediction over each time step on four Meta test sets

Table 8: Comparison of SDE scores over 24 steps ahead for consumption estimation

Model name	Test set 1	Test set 2	Test set 3	Test set 4
Best base Learner (Seq2Seq LSTM)	0.14	0.08	0.08	0.08
Averaged ensemble	0.21	0.16	0.07	0.07
Stacked Ensemble (Seq2Seq LSTM + Ridge)	0.09	0.13	007	0.07
Stacked Ensemble (Seq2Seq LSTM + GBRT)	**0.08**	**0.11**	**0.049**	**0.03**

Table 9: Comparison of SDE scores over 24 steps ahead for production estimation

Model name	Test set 1	Test set 2	Test set 3	Test set 4
Best base Learner (Seq2Seq LSTM)	0.2	**0.14**	**0.14**	**0.14**
Averaged ensemble	0.3	0.25	0.15	0.3
Stacked Ensemble (Seq2Seq LSTM + Ridge)	**0.18**	**0.14**	0.16	**0.14**
Stacked Ensemble (Seq2Seq LSTM + GBRT)	0.20	0.16	0.15	0.15

The degree of the overall variation in terms of MAE was computed by SDE metric. Results are presented in Table 8 and Table 9. It can be seen that the proposed ensemble show higher consistency in multi-step ahead forecasting of energy consumption across all test sets with on average 0.06 variation rate.

Regarding solar output estimations, higher variation values are recorded for all models. On average, the best base learners followed by the two ensembles reach satisfactory SDE scores of 0.15, 0.16. and 0.17, respectively. However, as expected, the Averaged Ensemble with high SDE values shows poor forecasting stability against the other techniques in most cases.

To further verify the superiority of the stacked models, the day-ahead (24 hours ahead) prediction results are analyzed under different situations. For load demand estimation, weekdays and weekends are considered whereas, for solar power output prediction, two typical weather conditions are evaluated: Cloudy and non-cloudy days. To identify cloudy days, we used the cloud visual opacity index, which describes how much sunlight the clouds let some sunlight through. The days with cloud opacity values less than 42 are considered as clear or partially cloudy days, while the days with higher index values are categorized as cloudy days.

Fig. 31 depicts the distribution of residual error of the proposed framework along with other comparative models regarding electricity consumption prediction of one community in Sydney. The residual error represents the difference between predicted and real values. As expected, the median error values of all models mostly increase during peak hours between 7.00 to 10.00 in the morning and from 17.00 to 21.00 in the afternoon. Moreover, for all frameworks, smaller boxes of weekdays compared to weekends indicate fewer variations in forecasting results and, therefore, more predictability of usage patterns in weekdays.

Forecast residual error comparison indicates that the stacked predictors generate less error in comparison with the best individual predictor and Averaged ensemble, demonstrating the potential benefit of the stacking ensemble technique in the day-ahead load demand forecast application.

Table 10 summarizes the prediction errors of different models in different communities. The proposed framework is more accurate than the comparative forecast models by producing at most 3.50 KW/h error for weekdays and 4.75 KW/h for weekends.

Table 10: Average residual error for day ahead forecasting of energy consumption based on type of day

	Test set 1		Test set 2		Test set 3		Test set 4	
Model/Day type	Weekday	Weekend	Weekday	Weekend	Weekday	Weekend	Weekday	Weekend
Best base learner (Seq2Seq LSTM)	3.81	5.2	3.48	4.97	3.24	3.38	3.68	3.98
Averaged ensemble	3.77	5.8	3.43	5.38	4.29	4.01	3.95	3.79
Stacked ensemble (Seq2Seq LSTM + Ridge)	3.51	4.34	3.33	**4.72**	3.15	3.38	3.60	3.80
Stacked ensemble (Seq2Seq LSTM + GBRT)	**3.50**	**4.27**	**3.28**	4.75	**3.10**	**3.29**	**3.25**	**3.56**

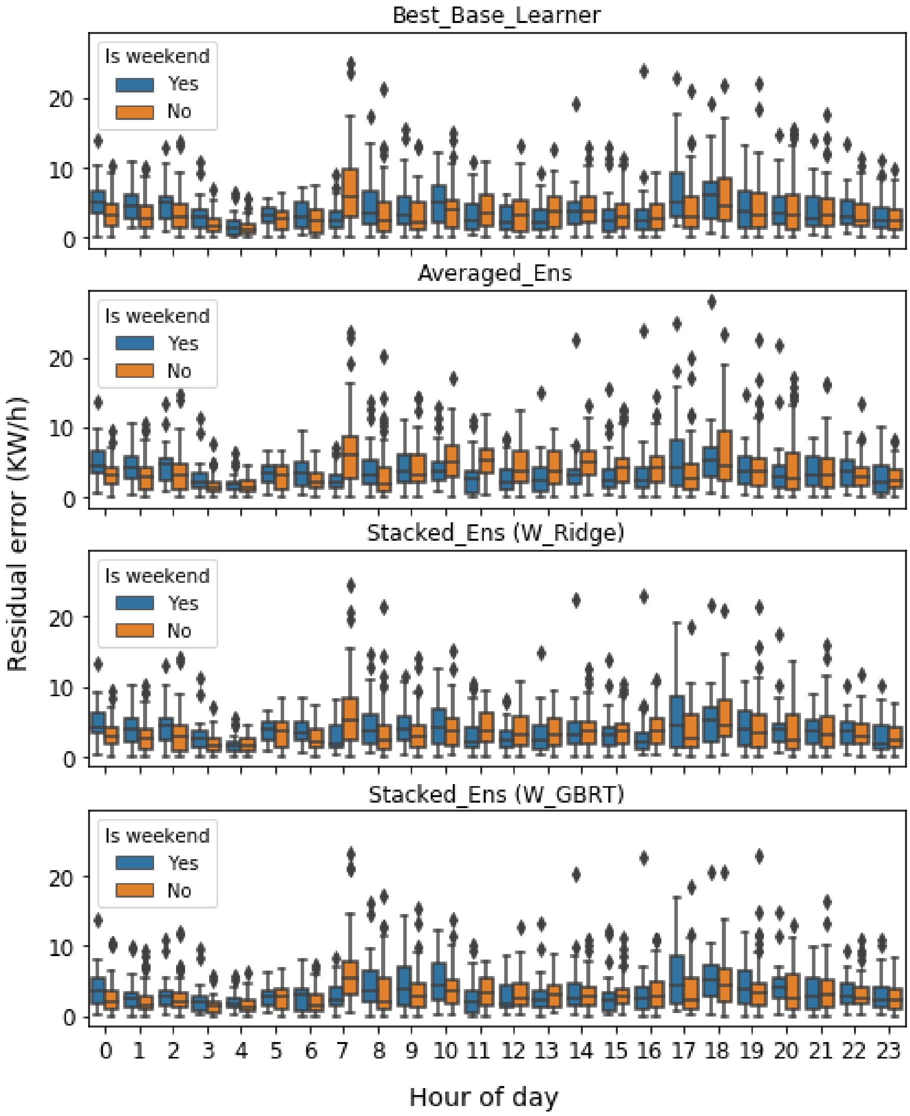

Figure 31: Error distribution of day-ahead load demand forecasting over weekdays and weekends

Fig. 32 highlights the box forecast error plot of all models for day-ahead estimation of PV power output in cloudy and non-cloudy days in the same test community.

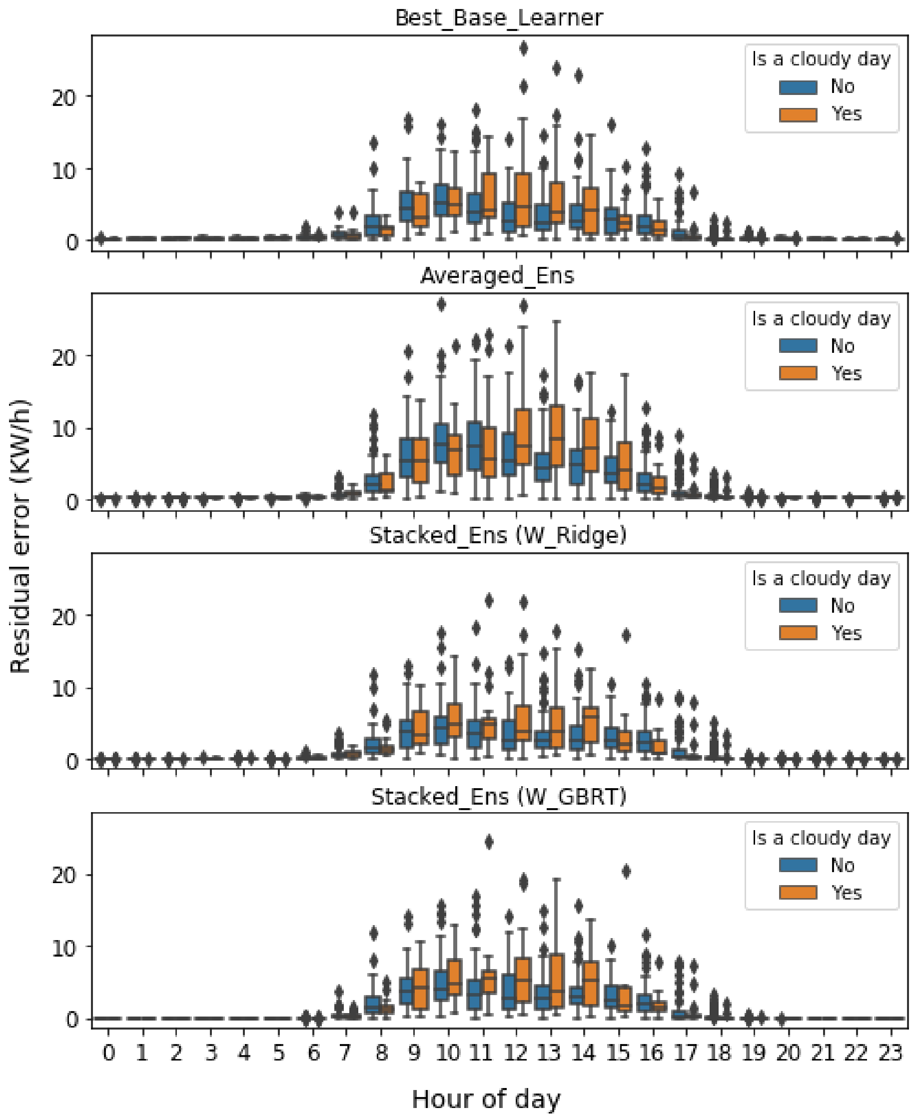

Figure 32: Error distribution of day-ahead solar output forecasting over cloudy and and non-cloudy days

It can be observed that the PV output prediction of each model is relatively less accurate on cloudy days, and this is primarily because the PV power curve on cloudy weather is less steady and more volatile. Compared to stacked models, the averaged ensemble predictor followed by the best learner gives higher forecast errors during cloudy days. Among all, the proposed ensemble with GBRT has produced the most accurate results with fewer outliers.

In Table 11, the results are summarized for all models and test sets. As shown, the proposed method can mostly capture the regular and non-regular patterns of PV power output at a satisfactory level on cloudy and non-cloudy days across different communities.

Table 11: Average residual error for day ahead forecasting of energy generation for cloudy and non-cloudy days

	Test set 1		Test set 2		Test set 3		Test set 4	
Model/Day type	Cloudy	Non-Cloudy	Cloudy	Non-Cloudy	Cloudy	Non-Cloudy	Cloudy	Non-Cloudy
Best base learner (Seq2Seq LSTM)	1.86	2.14	1.32	1.50	1.77	1.88	1.61	1.69
Averaged ensemble	2.08	2.41	1.63	1.80	3.32	4.94	2.34	2.36
Stacked ensemble (Seq2Seq LSTM + Ridge)	1.58	1.69	**1.08**	**1.30**	1.71	1.66	1.50	**1.41**
Stacked ensemble (Seq2Seq LSTM + GBRT)	**1.52**	**1.65**	**1.08**	1.38	**1.61**	**1.54**	**1.48**	1.43

4 Conclusion

In this paper, we proposed a framework for multi-hour ahead load forecasting and solar energy generation estimation of household communities. The framework introduces a process in which an ensemble model is developed based on extensive evaluations of baseline forecasting algorithms. The ensemble model applies deep recurrent neural networks as base learners and a tree-based ensemble algorithm as meta learner. It also incorporates multivariate time series data, including energy, time, and weather variables as predictive features to address the volatility of load series.

The proposed method offers several advantages over existing techniques. Firstly applying an ensemble learning strategy enables the model to provide more robust and accurate results than individual predictive methods. Secondly, deep recurrent neural networks as strong predictive algorithms for time series prediction tasks, provide the model with highly accurate base estimations. Next, since the ensemble model is not reliant on the structure of one particular deep network, it can generalize better to new data sets than individual neural networks that are heavily tuned for a given dataset. Finally, unlike the boosting approach that involves sequential learning, the applied stacking strategy offers the ability to separately train base learners, thereby reducing training time in distributed computational environments. However, the main limitation of the proposed approach can be the lack of appropriate historical data for proper training of deep base models.

In future work, the performance of the ensemble technique and all contributing algorithms can be evaluated through sensitivity analysis where we examine the impacts of different input features or input size on the prediction task. The presented forecast framework could also be applied for other types of time series data such as wind and electricity price as long as a sufficient amount of data (typically one to a few years of hourly observations) for training the deep networks are available.

References

[1] A. J. Waldau. "Snapshot of Photovoltaics." In: *Energies* 12.5 (Feb. 2019), p. 7.

[2] Ekanki Sharma. "Energy forecasting based on predictive data mining techniques in smart energy grids." In: *Energy Informatics* 1.1 (Oct. 2018), p. 44. ISSN: 2520-8942. DOI: 10.1186/s42162-018-0048-9.

[3] David Sembroiz, Davide Careglio, Sergio Ricciardi, and Ugo Fiore. "Planning and operational energy optimization solutions for smart buildings." In: *Information Sciences* 476 (2019), pp. 439–452. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2018.06.003.

[4] C. Wan, J. Zhao, Y. Song, Z. Xu, J. Lin, and Z. Hu. "Photovoltaic and solar power forecasting for smart grid energy management." In: *CSEE Journal of Power and Energy Systems* 1.4 (Dec. 2015), pp. 38–46.

[5] Zhenyu Wang, Jun Li, Shilin Zhu, Jun Zhao, Shuai Deng, Shengyuan Zhong, Hongmei Yin, Hao Li, Yan Qi, and Zhiyong Gan. "A Review of Load Forecasting of the Distributed Energy System." In: *IOP Conference Series: Earth and Environmental Science* 237 (Mar. 2019), p. 042019. DOI: 10.1088/1755-1315/237/4/042019.

[6] E. Sharma, M. Mussetta, and W. Elmenreich. "Investigating the impact of data quality on the energy yield forecast using data mining techniques." In: *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. 2020, pp. 599–603. DOI: 10.1109/ISGT-Europe47291.2020.9248920.

[7] Jiaqi Zhong, Luyao Liu, Qie Sun, and Xinyu Wang. "Prediction of Photovoltaic Power Generation Based on General Regression and Back Propagation Neural Network." In: *Energy Procedia* 152 (2018). Cleaner Energy for Cleaner Cities, pp. 1224–1229. ISSN: 1876-6102. DOI: https://doi.org/10.1016/j.egypro.2018.09.173.

[8] Luyao Liu, Yi Zhao, Dongliang Chang, Jiyang Xie, Zhanyu Ma, Qie Sun, Hongyi Yin, and Ronald Wennersten. "Prediction of short-term PV power output and uncertainty analysis." In: *Applied Energy* 228 (2018), pp. 700–711. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2018.06.112.

[9] Jinliang Zhang, Yi-Ming Wei, Dezhi Li, Zhongfu Tan, and Jianhua Zhou. "Short term electricity load forecasting using a hybrid model." In: *Energy* 158 (2018), pp. 774–781.

[10] Sobrina Sobri, Sam Koohi-Kamali, and Nasrudin Abd. Rahim. "Solar photovoltaic generation forecasting methods: A review." In: *Energy Conversion and Management* 156 (2018), pp. 459–497. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2017.11.019.

[11] S.L. Wong, Kevin K.W. Wan, and Tony N.T. Lam. "Artificial neural networks for energy analysis of office buildings with daylighting." In: *Applied Energy* 87.2 (2010), pp. 551–557. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2009.06.028.

[12] Betul Bektas Ekici and U. Teoman Aksoy. "Prediction of building energy consumption by using artificial neural networks." In: *Advances in Engineering Software* 40.5 (2009), pp. 356–362. ISSN: 0965-9978. DOI: https://doi.org/10.1016/j.advengsoft.2008.05.003.

[13] Ailing Yang, Weide Li, and Xuan Yang. "Short-term electricity load forecasting based on feature selection and Least Squares Support Vector Machines." In: *Knowledge-Based Systems* 163 (2019), pp. 159–173. ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2018.08.027.

[14] N. Singh, C. Vyjayanthi, and C. Modi. "Multi-step Short-term Electric Load Forecasting using 2D Convolutional Neural Networks." In: *2020 IEEE-HYDCON*. 2020, pp. 1–5. DOI: 10.1109/HYDCON48903.2020.9242917.

[15] Shamsul Masum, Y. Liu, and J. Chiverton. "Multi-step Time Series Forecasting of Electric Load Using Machine Learning Models." In: *ICAISC*. 2018.

[16] X. Shao and C. S. Kim. "Multi-Step Short-Term Power Consumption Forecasting Using Multi-Channel LSTM With Time Location Considering Customer Behavior." In: *IEEE Access* 8 (2020), pp. 125263–125273. DOI: 10.1109/ACCESS.2020.3007163.

[17] Yi Yang, Zhihao Shang, Yao Chen, and Yanhua Chen. "Multi-Objective Particle Swarm Optimization Algorithm for Multi-Step Electric Load Forecasting." In: *Energies* 13.3 (2020). ISSN: 1996-1073. DOI: 10.3390/en13030532.

[18] Weide Li, Xuan Yang, Hao Li, and Lili Su. "Hybrid Forecasting Approach Based on GRNN Neural Network and SVR Machine for Electricity Demand Forecasting." In: *Energies* 10.1 (2017). ISSN: 1996-1073.

[19] J. Zhang, Y. Chi, and L. Xiao. "Solar Power Generation Forecast Based on LSTM." In: *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*. 2018, pp. 869–872. DOI: 10.1109/ICSESS.2018.8663788.

[20] André Gensler, Janosch Henze, Bernhard Sick, and Nils Raabe. "Deep Learning for solar power forecasting—An approach using AutoEncoder and LSTM Neural Networks." In: *2016 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE. 2016, pp. 002858–002865.

[21] HyunYong Lee and Byung-Tak Lee. "Bayesian Deep Learning-based Confidence-aware Solar Irradiance Forecasting System." In: *2018 International Conference on Information and Communication Technology Convergence (ICTC)* (2018), pp. 1233–1238.

[22] Ahmad Alzahrani, Pourya Shamsi, Cihan Dagli, and Mehdi Ferdowsi. "Solar Irradiance Forecasting Using Deep Neural Networks." In: *Procedia Computer Science* 114 (2017). Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA, pp. 304–313. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2017.09.045.

[23] Zibo Dong, Dazhi Yang, Thomas Reindl, and Wilfred M. Walsh. "A novel hybrid approach based on self-organizing maps, support vector regression and particle swarm optimization to forecast solar irradiance." In: *Energy* 82 (2015), pp. 570–577. ISSN: 0360-5442. DOI: `https://doi.org/10.1016/j.energy.2015.01.066`.

[24] Javier Antonanzas, Ruben Urraca, Alpha Pernía-Espinoza, Alvaro Aldama, Luis Alfredo Fernández-Jiménez, and Francisco Javier Martínez-de-Pisón. "Single and Blended Models for Day-Ahead Photovoltaic Power Forecasting." In: *Hybrid Artificial Intelligent Systems*. Ed. by Francisco Javier Martínez de Pisón, Rubén Urraca, Héctor Quintián, and Emilio Corchado. Cham: Springer International Publishing, 2017, pp. 427–434.

[25] Muhammad Waseem Ahmad, Monjur Mourshed, and Yacine Rezgui. "Tree-based ensemble methods for predicting PV power generation and their comparison with support vector regression." In: *Energy* 164 (2018), pp. 465–474. ISSN: 0360-5442. DOI: `https://doi.org/10.1016/j.energy.2018.08.207`.

[26] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: data mining, inference, and prediction*. Springer, New York, NY, 2009. DOI: `https://doi.org/10.1007/978-0-387-84858-7`.

[27] Yang Liu, Wei Wang, and Noradin Ghadimi. "Electricity load forecasting by an improved forecast engine for building level consumers." In: *Energy* 139 (2017), pp. 18–30. ISSN: 0360-5442. DOI: `https://doi.org/10.1016/j.energy.2017.07.150`.

[28] Woohyun Kim, Yerim Han, Kyoung Jae Kim, and Kwan-Woo Song. "Electricity load forecasting using advanced feature selection and optimal deep learning model for the variable refrigerant flow systems." In: *Energy Reports* 6 (2020), pp. 2604–2618. ISSN: 2352-4847. DOI: `https://doi.org/10.1016/j.egyr.2020.09.019`.

[29] Cheng Fan, Fu Xiao, and Shengwei Wang. "Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques." In:

Applied Energy 127 (2014), pp. 1–10. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2014.04.016.

[30] A. M. Pirbazari, A. Chakravorty, and C. Rong. “Evaluating Feature Selection Methods for Short-Term Load Forecasting.” In: *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2019, pp. 1–8. DOI: 10.1109/BIGCOMP.2019.8679188.

[31] Nasir Ayub, Nadeem Javaid, Sana Mujeeb, Maheen Zahid, Wazir Zada Khan, and Muhammad Umar Khattak. “Electricity Load Forecasting in Smart Grids Using Support Vector Machine.” In: *Advanced Information Networking and Applications*. Ed. by Leonard Barolli, Makoto Takizawa, Fatos Xhafa, and Tomoya Enokido. Springer International Publishing, 2020.

[32] James Durbin and Siem Jan Koopman. *Time series analysis by state space methods*. Oxford university press, 2012.

[33] G Peter Zhang. “Time series forecasting using a hybrid ARIMA and neural network model.” In: *Neurocomputing* 50 (2003), pp. 159–175.

[34] Hongzhan Nie, Guohui Liu, Xiaoman Liu, and Yong Wang. “Hybrid of ARIMA and SVMs for short-term load forecasting.” In: *Energy Procedia* 16 (2012), pp. 1455–1460.

[35] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*. Vol. 26. Springer, 2013.

[36] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[37] Wei-Chiang Hong. “Electric load forecasting by support vector model.” In: *Applied Mathematical Modelling* 33.5 (2009), pp. 2444–2454.

[38] Zhenqi Tan, Jing Zhang, Yu He, Ying Zhang, Guojiang Xiong, and Ying Liu. “Short-term Load Forecasting based on Integration of SVR and stacking.” In: *IEEE Access* (2020).

[39] Mohamed Abuella and Badrul Chowdhury. *Solar Power Forecasting Using Support Vector Regression*. 2017. arXiv: 1703.09851 [cs.LG].

[40] Oliver Kramer and Fabian Gieseke. "Short-Term Wind Energy Forecasting Using Support Vector Regression." In: *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*. Ed. by Emilio Corchado, Václav Snášel, Javier Sedano, Aboul Ella Hassanien, José Luis Calvo, and Dominik Ślzak. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 271–280.

[41] Y. Zhang, H. Sun, and Y. Guo. "Wind Power Prediction Based on PSO-SVR and Grey Combination Model." In: *IEEE Access* 7 (2019), pp. 136254–136267. DOI: 10.1109/ACCESS.2019.2942012.

[42] Devon K Barrow and Sven F Crone. "A comparison of AdaBoost algorithms for time series forecast combination." In: *International Journal of Forecasting* 32.4 (2016), pp. 1103–1119.

[43] Jerome H Friedman. "Stochastic gradient boosting." In: *Computational Statistics & Data Analysis* 38.4 (2002), pp. 367–378.

[44] Jing Li, Ji-hang Cheng, Jing-yuan Shi, and Fei Huang. "Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement." In: *Advances in Computer Science and Information Engineering*. Ed. by David Jin and Sally Lin. Springer Berlin Heidelberg, 2012, pp. 553–558.

[45] A.S. Ahmad, M.Y. Hassan, M.P. Abdullah, H.A. Rahman, F. Hussin, H. Abdullah, and R. Saidur. "A review on applications of ANN and SVM for building electrical energy consumption forecasting." In: *Renewable and Sustainable Energy Reviews* 33 (2014), pp. 102–109. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2014.01.069. URL: http://www.sciencedirect.com/science/article/pii/S1364032114000914.

[46] Utpal Kumar Das, Kok Soon Tey, Mehdi Seyedmahmoudian, Saad Mekhilef, Moh Yamani Idna Idris, Willem Van Deventer, Bend Horan, and Alex Stojcevski. "Forecasting of photovoltaic power generation and model optimization: A review." In: *Renewable and Sustainable Energy Reviews* 81 (2018), pp. 912–928. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2017.08.017. URL: http://www.sciencedirect.com/science/article/pii/S1364032117311620.

[47] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[48] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019, pp. 497–514.

[49] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. "On the properties of neural machine translation: Encoder-decoder approaches." In: *arXiv preprint arXiv:1409.1259* (2014).

[50] Muhammad Sajjad, Zulfiqar Ahmad Khan, Amin Ullah, Tanveer Hussain, Waseem Ullah, Mi Young Lee, and Sung Wook Baik. "A novel CNN-GRU-based hybrid approach for short-term residential load forecasting." In: *IEEE Access* 8 (2020), pp. 143759–143768.

[51] Yusen Wang, Wenlong Liao, and Yuqing Chang. "Gated recurrent unit network-based short-term photovoltaic forecasting." In: *Energies* 11.8 (2018), p. 2163.

[52] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. "Convolutional neural networks for speech recognition." In: *IEEE/ACM Transactions on audio, speech, and language processing* 22.10 (2014), pp. 1533–1545.

[53] Mahbub Hussain, Jordan J. Bird, and Diego R. Faria. "A Study on CNN Transfer Learning for Image Classification." In: *Advances in Computational Intelligence Systems*. Ed. by Ahmad Lotfi, Hamid Bouchachia, Alexander Gegov, Caroline

Langensiepen, and Martin McGinnity. Springer International Publishing, 2019, pp. 191–202.

[54] Irena Koprinska, Dengsong Wu, and Zheng Wang. "Convolutional neural networks for energy time series forecasting." In: *2018 international joint conference on neural networks (IJCNN)*. IEEE. 2018, pp. 1–8.

[55] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks." In: *arXiv preprint arXiv:1409.3215* (2014).

[56] J Scott Armstrong. "Combining forecasts." In: *Principles of forecasting*. Springer, 2001, pp. 417–439.

[57] Sascha Krstanovic and Heiko Paulheim. "Ensembles of recurrent neural networks for robust time series forecasting." In: *International Conference on Innovative Techniques and Applications of Artificial Intelligence*. Springer. 2017, pp. 34–46.

[58] *Ausgrid, the largest distributor of electricity on Australia's east coast*. URL: `https://www.ausgrid.com.au/Industry/Our-Research/Data-to-share/Solar-home-electricity-data` (visited on 12/01/2013).

[59] *Australian Government Bureau of Meteorology*. URL: `http://www.bom.gov.au/` (visited on 12/01/2020).

[60] V. Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines." In: *ICML*. 2010.

[61] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017.